

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: N2612 Elektrotechnika a informatika

Studijní obor: 1802T007 Informační technologie

Rozšiřování systému Delmia Process Engineer v oblasti plánování montáže automobilů

Upgrading of the Delmia Process Engineer for automotive assembly planning

Diplomová práce

Autor: **Bc. Zdeněk Šolc**

Vedoucí práce: Ing. Martin Vích Vlasák

Konzultant: Ing. Michael Woide

V Liberci 18 5. 2012

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Akademický rok: **2011/12**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Zdeněk Šolc**
Osobní číslo: M08000236
Studijní program: N2612 Elektrotechnika a informatika
Studijní obor: Informační technologie
Název tématu: Rozšiřování systému Delmia Process Engineer v oblasti plánování montáže automobilů

Zadávací katedra: Ústav mechatroniky a technické informatiky

Zásady pro vypracování:

1. Seznamte se s pojmem „Digitální továrna“
2. Seznamte se s programovým balíkem Delmia Process Engineer a s možnostmi jeho rozšíření a přizpůsobení po konkrétní průmyslovou aplikaci. Jak z pohledu uživatele, tak hlavně z pohledu programátora.
3. Implementujte, otestujte a zdokumentujte automatické změny časové platnosti dílu, přiřazené k operaci.
4. Implementujte, otestujte a zdokumentujte tvorbu maker pro inteligentní regály.

Rozsah grafických prací: **dle potřeby dokumentace**

Rozsah průvodní zprávy: **cca 40-50 stran**

Forma zpracování diplomové práce: **tištěná / elektronická**

Seznam odborné literatury:

[1] Dokumentace k systému Delmia Process Engineer

Vedoucí diplomové práce: **Ing. Martin Vích Vlasák**

Ústav mechatroniky a technické informatiky

Konzultant diplomové práce: **Ing. Michael Woide**

firma T-Systems Czech Republic a.s.

Datum zadání diplomové práce: **12. září 2011**

Datum odevzdání diplomové práce: **18. května 2012**

prof. Ing. Václav Kopecký, CSc.
děkan

doc. Ing. Petr Tůma, CSc.
vedoucí ústavu

V Liberci dne 12. září 2011

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum: 18. 5. 2012

Podpis

Poděkování

Rád bych poděkoval panu Ing. Martinu Vích Vlasákovi za odborné vedení a podnětné připomínky a panu Ing. Michaelu Woidemu za návrh tématu této práce a cenné rady při zpracování.

Dále bych chtěl poděkovat příteli a kolegovi Pavlu Jablonskému za poradenství v oblasti programování a programovacích technik.

A v neposlední řadě patří můj dík i všem pedagogům a akademikům, kteří mi během mého studia trpělivě předávali své zkušenosti a znalosti.

Abstrakt

Tato práce pojednává o digitální továrně, jejích možnostech a nástrojích, které jsou v současné době k dispozici, zároveň popisuje možnosti rozšiřování těchto nástrojů a konkrétní úpravy vybraného systému dle přání zákazníka.

Je zde popsán pojem „digitální továrna“, její výhody a nevýhody. Dále jsou zde zmíněni výrobci a jejich systémy používané pro plánování a simulaci výroby. Vybrán je systém Delmia Process Engineer, který je popsán podrobněji. Je zde uvedeno, jakým způsobem pracuje a jak ho lze modifikovat a rozšiřovat. Následně je tento systém upravován dle přání zákazníka. Tyto úpravy obsahují dvě úlohy, které mají usnadnit práci uživatelům, kteří se systémem pracují. Tato rozšíření jsou vypracována pomocí naprogramování nových skriptů a vývojem samostatné aplikace, která je následně převedena na dll knihovnu. Tato knihovna je pak implementována do systému.

Tyto úpravy jsou vypracovány do podrobných detailů a otestovány pro následné užití při návrhu sériové výroby.

Klíčová slova

Delmia Process Engineer

Digitální továrna

MTM analýza

Životní cyklus výrobku

Abstract

This work describes a digital factory, its abilities and options that are currently available. It also elaborates possible versatility of these options and customer-driven modifications.

A new term "digital factory" is defined in the work and possible advantages and disadvantages of such a factory are listed. Producers and their systems for production planning and simulation are mentioned. System Delmia Process Engineer is described in details. Its default principles are mapped together with possible extensions and modifications. The system is then modified according to customer demands. The modification consists of two parts that are both designed to make the system more user-friendly. New scripts and further developments are transformed into a dll library. The library is consequently implemented back to the system.

The modifications are designed in details and tested for subsequent use in a serial production development.

Key words

Delmia Process Engineer

Digital factory

Methods-Time Measurement

Product Lifecycle Management

Obsah

Prohlášení.....	3
Poděkování.....	4
Abstrakt.....	5
Abstract	6
Seznam obrázků.....	9
Seznam náhledů do zdrojových kódů	10
Seznam tabulek.....	10
Seznam zkratk.....	11
Úvod	12
1. Digitální továrna.....	14
1.1. PLM.....	15
1.2. Hlavní přínosy digitální továrny	15
1.3. Firmy používající nástroje DF	18
1.4. Systémy pro digitální továrnu	18
2. Delmia Process Engineer	19
2.1. DPE z programátorského hlediska	22
2.2. Customizace DPE.....	24
2.2.1. Visual Basic Script.....	25
2.2.2. Visual Basic .NET	27
2.2.3. Konfigurační modul.....	28
2.2.4. Propojení DPE s dalšími systémy	29
3. Úloha 1.....	30
3.1. Zadání.....	30
3.1.1. Současný stav	30
3.1.2. Požadovaný stav	31
3.2. Analýza	32
3.3. Řešení.....	35
3.4. Testování	38
4. Úloha 2.....	40
4.1. Zadání.....	40
4.1.1. Obecný popis	40
4.1.2. Používaná úložiště	41

4.1.3. Požadavky na uživatelské prostředí.....	42
4.1.4. Požadované funkcionality a chování aplikace	43
4.2. Analýza	44
4.2.1. Příprava a načtení dat.....	44
4.2.2. Nová implementovaná aplikace - vzhled.....	46
4.2.3. Nová implementovaná aplikace – dialogová okna	46
4.2.4. Návrat do DPE	46
4.2.5. Přepřavky	47
4.3. Řešení.....	47
4.3.1. Skripty upravující data pro implementovanou aplikaci.....	47
4.3.2. Implementovaná aplikace	49
4.3.3. Skript zapisující data do DPE	56
4.3.4. Implementace do systému.....	57
4.4. Materiály pro školení	58
4.5. Testování	58
4.6. Navázání na tuto úlohu.....	59
Závěr.....	60
Použitá Literatura	62
Seznam příloh	64

Seznam obrázků

Obr. 1-1: Graf závislosti nákladů na změnách v průběhu plánování (autor).....	14
Obr. 1-2: V5 Ergonomics - zatížení člověka (Digitální fabrika – výukové materiály FS, ZČU).....	18
Obr. 2-1: Prostředí Delmia Process Engineer (autor).....	19
Obr. 2-2: PPR strom v DPE (autor).....	20
Obr. 2-3: Schéma DPE (autor).....	22
Obr. 2-4: Schéma PPR - API (autor)	24
Obr. 2-5: Spouštění skriptového příkazu ze seznamu a z kontextového menu (autor).....	26
Obr. 2-6: Okno s nastavováním spouštění skriptové akce (autor).....	27
Obr. 2-7: Okno konfiguračního modulu (autor).....	29
Obr. 3-1: Schéma současného stavu (autor).....	31
Obr. 3-2: Schéma požadovaného stavu (autor)	32
Obr. 3-3: Schéma struktury vazby díl-operace (autor)	32
Obr. 3-4: Schéma skriptového příkazu (autor)	34
Obr. 3-5: Schéma skriptové akce (autor)	35
Obr. 3-6: Popis funkce sa_link (User Manual DPE)	37
Obr. 3-7: Znázornění platnosti dílu (autor).....	39
Obr. 3-8: Znázornění platnosti vazby (díl-operace) (autor)	39
Obr. 4-1: Schéma propojení původní aplikace s DPE.....	40
Obr. 4-2: Standardní regál 0-8-0-0-6 (autor).....	41
Obr. 4-3: Diagram možností uložení přepravek v regálu (autor)	42
Obr. 4-4: Vzhled původní aplikace (autor).....	42
Obr. 4-5: Schéma požadovaných funkcí	43
Obr. 4-6: Náskres linky (autor).....	44
Obr. 4-7: Struktura výrobní linky (autor).....	45
Obr. 4-8: Struktura výrobní linky s novým uzlem (autor)	45
Obr. 4-9: Tři hlavní části úlohy (autor)	47
Obr. 4-10: Funkce skriptů pro přípravu dat (autor)	48
Obr. 4-11: Schéma načítání struktury úložišť	48
Obr. 4-12: Diagram hlavních funkcí aplikace (autor)	50
Obr. 4-13: Hlavní formulář implementované aplikace (autor)	53
Obr. 4-14: Náhled na okno s ukládáním KLT- přepravek (autor)	54
Obr. 4-15: Schéma plnění regálu (autor)	55
Obr. 4-16: Nová generovaná struktura úložišť (autor).....	57

Seznam náhledů do zdrojových kódů

Náhled č. 2-1: Příklad zdrojového kódu ve VBS – vypsaní plánovacího typu objektu	25
Náhled č. 3-1: Ukázka ze skriptu č. 1 – načtení aktuálního data a nastavení prvního období.	36
Náhled č. 3-2: Ukázka ze skriptu č. 1 – query	36
Náhled č. 4-1: Skript v DPE odkazující na registrovanou knihovnu dll	58

Seznam tabulek

Tabulka č. 1: Rozměry používaných přepravek	41
Tabulka č. 2: Rozměry používaných přihrádek v regálu.....	41

Seznam zkratek

AFO – Arbeitsfolge (krok operace)

ALB – Automatic line balancing (automatické taktování linky)

API – Application programming interface (Aplikační programové rozhraní)

DF – Digital Factory (Digitální továrna)

DLL - Dynamic Linked Library (dynamicky linkovaná knihovna)

DPE - Delmia Process Engineer

DS – Dassault Systèmes

GLT – Grossladungsträger (velká standardní přepravka)

KLT – Kleinladungsträger (malá standardní přepravka)

MTM - Methods-Time Measurement

PLM - Product Lifecycle Management (Životní cyklus výrobku)

VB.NET - Visual Basic .Net

VBS - Visual Basic Script

Úvod

Diplomová práce pojednává o použití systému Delmia Process Engineer v oblasti plánování montáže automobilů.

Tato diplomová práce byla navržena společností T-Systems Czech Republic a.s., která je v české republice dodavatelem těchto systémů a zároveň provozuje i jejich uživatelskou podporu.

Jsou zde zmíněny základní informace o digitální továrně včetně výhod a nevýhod používání. Dále pak seznámení s nejdůležitějšími systémy pro práci s digitální továrnou, kde je jeden vybrán a popsán podrobněji. Zároveň je zde část, která je zaměřena na úpravy tohoto systému.

Vybrán je systém Delmia Process Engineer od společnosti Dassault Systemes. Jde o databázový systém, který pracuje v oblasti digitální továrny a je využíván pro kompletní plánování sériové výroby.

Protože byl tento systém následně upravován, byl zde popsán především z programátorského hlediska. Byly zde popsány zejména způsoby jeho rozšiřování a upravování.

Další část je zaměřena na konkrétní úlohy týkající se rozšíření systému Delmia Process Engineer. Tyto úlohy jsou podrobně popsány. Počínaje zadáním, vypracováním analýzy a řešení, které popisuje kompletní vypracování úloh.

První úprava systému obsahuje automatickou změnu dat dle aktuálního období, která je řešena pomocí skriptů (kapitola 3).

Druhá úloha má umožnit uživatelům jednoduchou práci s přiřazováním jednotlivých přepravek do regálů na konkrétních stanovištích výrobní linky (kapitola 4). Toto rozšíření je založeno na skriptech a samostatně vyvíjené aplikaci, ze které je následně odvozena knihovna dll.

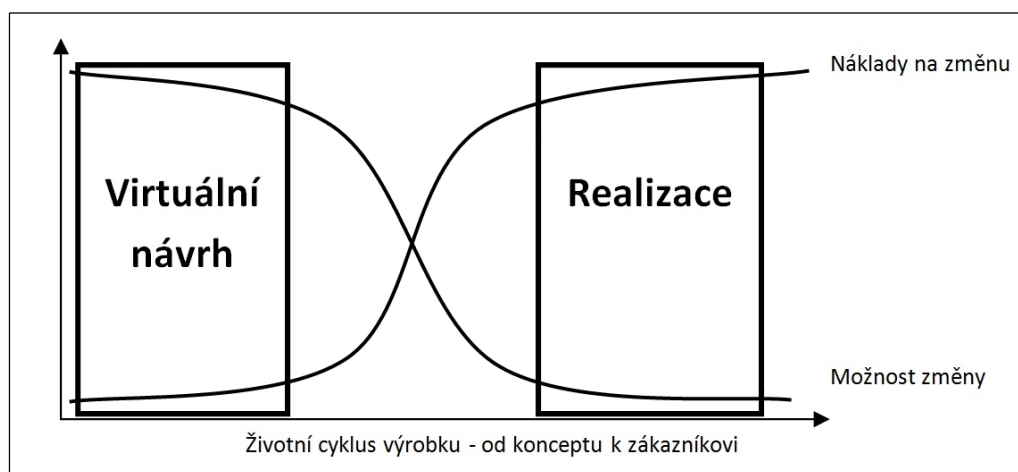
Realizace těchto úloh je doprovázena závěrečným testováním před uvedením na produkční server.

Po zhotovení této práce byl vypracován i návrh na pokračování. Jde o Vytvoření 3D vizualizace montážního prostředí pomocí dat z druhé úlohy.

1. Digitální továrna

Pojmem „digitální továrna“ (dále jen DF podle „Digital Factory“) je definován jako rozsáhlý souhrn digitálních metod, modelů a nástrojů (včetně simulace a 3D vizualizace). Jejich cílem je komplexní a systémové plánování, projektování, ověřování a následně průběžné zlepšování všech důležitých struktur, procesů a zdrojů reálné továrny. Digitální nástroje jsou navrhovány pro řízení celého životního cyklu výrobku (PLM).

Softwarové nástroje umožňují zavádět digitální továrnu do praxe. Pomocí přesného modelování, simulací a 3D vizualizací mohou všichni, kteří kooperují během vývoje, zobrazovat a analyzovat budoucí výrobní procesy. Tato vyhodnocení umožňují, aby zásadní rozhodnutí byla provedena včas. Předěje se tak možným chybám, které by se jinak mohly objevit až při spuštění výroby. Digitalizace umožňuje rychlejší a pečlivější přípravu procesů. Simulace a optimalizace v návrhové fázi zajistí, aby bylo dosaženo kompletní a kvalitní výroby produktu bez jakýchkoli chyb hned při prvním náběhu výroby. Toto zamezí dodatečným a zbytečným velmi vysokým nákladům v reálné továrně (Obr. 1-1). [2]



Obr. 1-1: Graf závislosti nákladů na změnách v průběhu plánování (autor)

Výhodou je možnost včasného zohlednění požadavků výroby a montáže již ve fázi vývoje výrobku a také zajištění včasného zachycení prvotních plánovacích úvah a analýz do výsledků plánovacích procesů.

Jelikož jsou dnes výrobci pod tlakem zákazníků, kteří je nutí k produkci nových výrobků ve stále kratší době, za nižší cenu i při zachování nejvyšší kvality, musí se poohlížet po možných nástrojích, které by jim pomohly tomuto nátlaku čelit. To platí ve všech oblastech průmyslové výroby. Proto je důležité klást velký důraz na zkrácení doby vývoje a přípravy sériové výroby.

Digitální továrna je jedna z nejvíce rozvíjejících se částí PLM.

1.1. PLM

PLM = Product Lifecycle Management zahrnuje všechny druhy plánování výroby, jako jsou:

- **Vývoj výrobku** - design a struktura, konstrukce a výpočty
- **Plánování** - procesní vývoj, technologie, výrobní plánování, toky materiálu
- **Projektování** - tvorba procesu, prostorová struktura, automatizace, konstrukce pomocných zařízení a přípravků
- **Vizualizace procesů** - materiálová, ergonomická kinematická simulace, programování, validace
- **Reálné zařízení** - výrobní a montážní stroje a zařízení, výkonové testy, výroba (montáž)

1.2. Hlavní přínosy digitální továrny

- minimalizování rizik při zavádění nové výroby
- ověřování návrhu konceptu výroby
- simulace výrobních linek, robotů a pracovních postupů
- možnost virtuální prohlídky výrobních hal
- prověření procesů před zahájením výroby
- zhodnocení ergonomie pracovišť
- vymezení potřebné plochy a úprav zařízení
- odhalení slabých míst a kolizí
- programování strojů a linek off-line
- omezení potřeby prototypů
- Pro firmy, které již DF zavádějí, mají velké přednosti v konkurenceschopnosti.

- Čas – zrychlení všech procesů
- Kvalita – bezproblémová výroba
- Náklady- zamezení nákupu zbytečného materiálu
- Uvedení na trh (flexibilita) – rychlá reakce na požadavky klienta

Čas (až 15%)

Navrhování a analyzování probíhá ve virtuálním světě rychleji, než ve světě reálném

- Zrychlení vývoje

Kvalita (až 15%)

S využitím podpůrných programů a simulací se zvyšuje míra vyhledání a odstranění chyb.

- Optimalizace logistiky zásobování polotovary a subdodávkami, nástrojového vybavení, průtoků atp.

Náklady (na výrobní zařízení až 40%)

- Zamezení vzniku duplicit v průběhu přípravných etap
- Optimalizace plánování a projektování výroby (pokles investičních potřeb, vyvážení využití, zvýšení průtoků výrobou)
- Snížení zmetkovitosti

Uvedení na trh (urychlení až o 20%)

- Podstatné zkrácení přípravných etap
- Zrychlení a zkvalitnění náběhu výroby (proběhly simulace a zkoušky na virtuálním modelu pracoviště)

Digitální továrna je stále ještě novou záležitostí (ověřenou pouze u největších koncernů), proto se při jejím používání mohou naskytnout problémy. Stále je co upravovat a vyvíjet ve všech modulech, které jsou schopny ulehčit práci průmyslovým inženýrům a technologům.

Oficiální výsledky přínosů DF

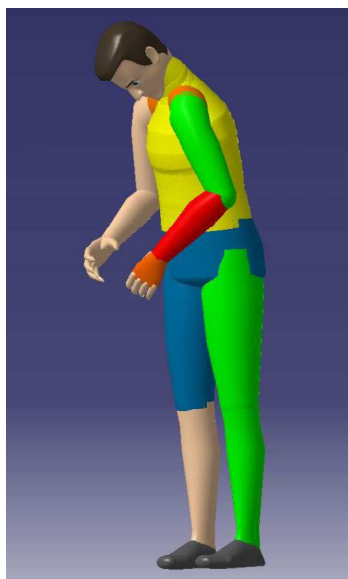
Úspory nákladů díky snížení majetku:	10%
Úspora ploch díky optimalizaci layoutů:	25%
Úspora nákladů díky lepšímu využití zdrojů:	30%
Úspora nákladů díky optimalizaci materiálových toků:	35%
Snížení počtu strojů, nástrojů a pracovišť:	40%
Celkové snížení nákladů:	13%
Celkové zvýšení produkce:	15%
Zkrácení doby nutné k uvedení výrobku na trh:	30%

[2] (Zdroj: The Benefits of Digital Manufacturing, CIMdata report)

Nevýhody DF

- Jejich cena je pro spoustu firem nepřijatelná, proto je používají především velké podniky se sériovou výrobou nebo s drahými výrobky.
- Kvůli velkému množství požadavků od uživatelů jsou tyto systémy stále ještě vyvíjeny.

Každý proces musí být navržen tak, aby mohl montážní dělník tento proces provádět co možná nejrychleji. Zároveň je nutné dbát i na ergonomii, která je důležitá jak pro rychlost provádění úkonů, tak i pro eliminaci zatížení dotyčného dělníka (Obr. 1-2).



Obr. 1-2: V5 Ergonomics - zatížení člověka (Digitální fabrika – výukové materiály FS, ZČU)

1.3. Firmy používající nástroje DF

Mezi nejvýznamnější uživatele systémů DF patří například společnosti: Airbus, Audi, BMW, Boeing, DaimlerChrysler, Fiat, Ford, GM, Kia Motors, Lockheed Martin, Mazda, Opel, Peugeot, Renault, Saab, Siemens, Škoda Auto, Toyota, Volkswagen a další.

1.4. Systémy pro digitální továrnu

Všichni výše zmiňovaní výrobci používají jeden z těchto dvou systémů nebo kombinaci jejich modulů od těchto výrobců.

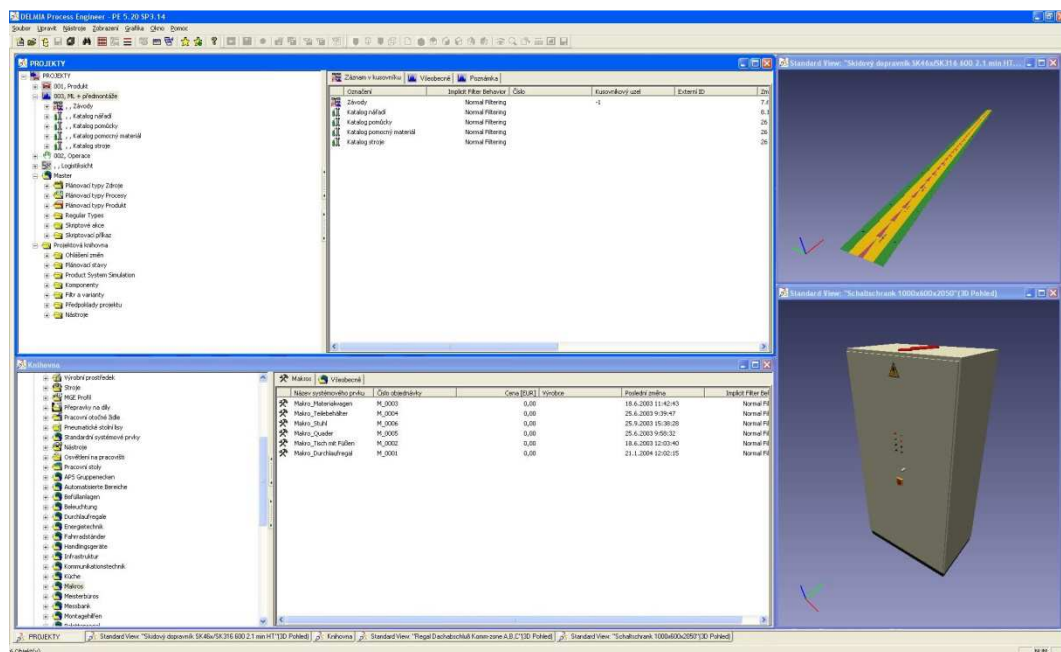
1. **Siemens** – řada **Tecnomatix** - nástroj **Process Designer**
2. **Dassault Systemes** - řada **Delmia** - nástroj **Delmia Process Engineer**

Tyto dvě řady systémů zastřešují všechny části výroby, jako jsou například veškeré procesy, které jsou zapotřebí k výrobě produktu. 3D návrhy výrobních linek, včetně simulací montážních pracovišť, ergonomie člověka i robota na pracovišti a dokonce i logistiku všech dílů potřebných k výrobě (včetně paletizací a ukládání do regálů na pracovišti i ve skladu).

Pro plánování montáže se dá použít několik dalších programů, ale již žádný jiný výrobce se nezabývá takovými důkladnými a kompletními nástroji.

2. Delmia Process Engineer

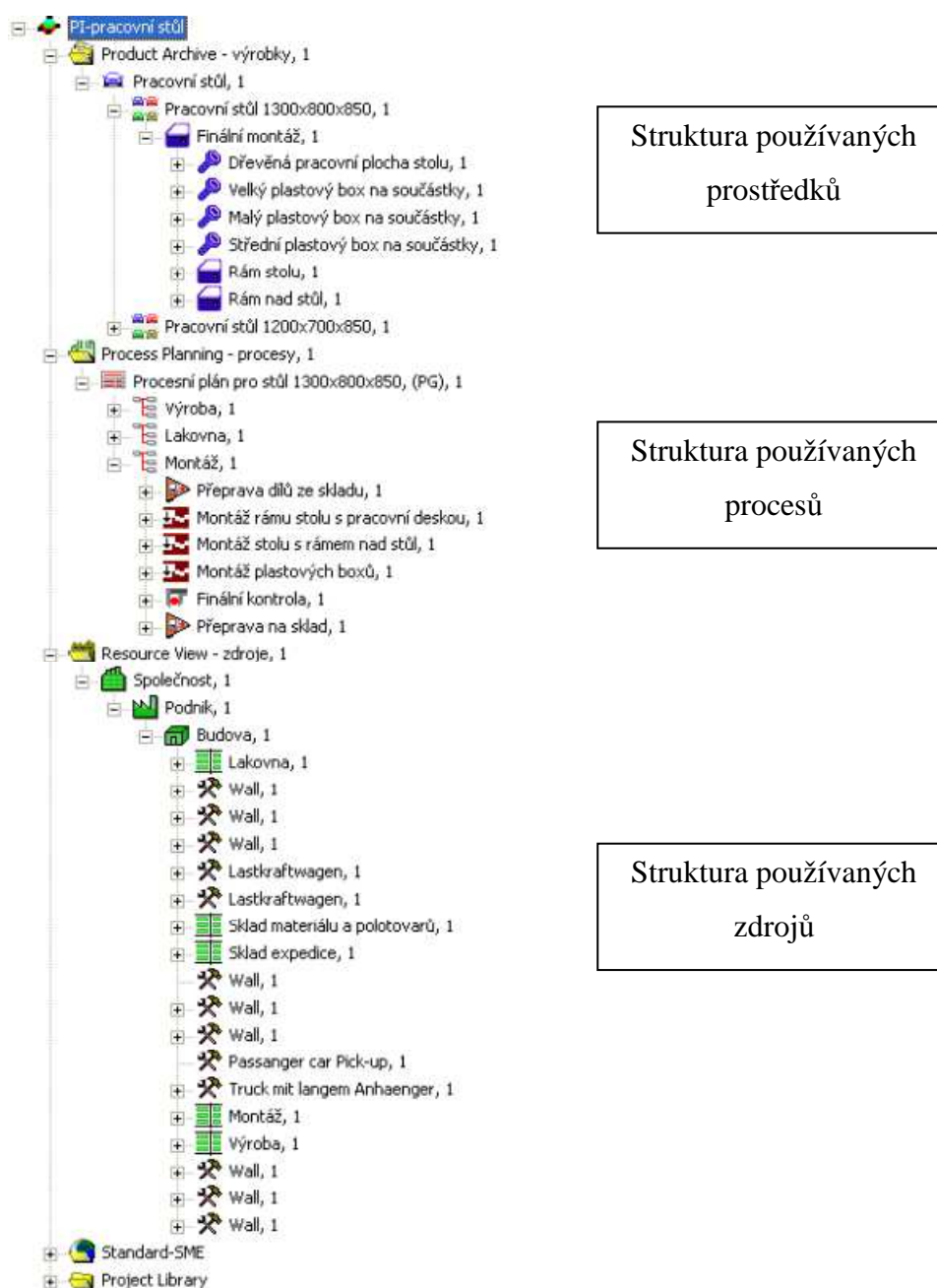
Delmia Process Engineer (dále jen DPE) je softwarový nástroj pro plánování výrobních procesů hromadné a sériové výroby a výroby složitých výrobků (Obr. 2-1).



Obr. 2-1: Prostředí Delmia Process Engineer

(autor)

Jedná se o produkt koncernu Dassault Systèmes Group, který je předním světovým dodavatelem řešení pro oblast PLM. Produkty společnosti DELMIA společně s dalšími koncernovými produkty (SOLIDWORKS, CATIA, ENOVIA, SIMULIA) vynikají díky svému komplexnímu přístupu k řešení problematiky v přípravě výroby a díky možnosti vzájemné integrace pomocí technologie **PPR Hub** (Product – Process – Resource).



Obr. 2-2: PPR strom v DPE

(autor)

Systém DPE je v současné době dodáván na platformě E5. Pro pokročilé činnosti přípravy výroby např. 3D simulace výrobních procesů, ergonomické studie, simulace materiálových toků apod. je nutné systém propojit s moduly platformy V5. Data z DPE (popis výrobního procesu nebo jeho části) slouží jako vstupní data těmto pokročilým modulům.

Nabízená řešení a jednotlivé produkty DELMIA je možno rozdělit do pěti hlavních oblastí:

Operace

Delmia operations: sada ověřených škálovatelných správ a realizací řešení, která umožňují výrobní společnosti optimalizovat procesy výroby a kvalitu výrobku.

Tovární definice a simulace.

Delmia robotics: škálovatelné řešení, které využívá výkon výrobního hubu k poskytování integrovaného, škálovatelného, flexibilního a snadno použitelného řešení, jako DELMIA Human, která používá sadu lidské simulace.

Digitální výrobní základy

Založené na převratných technologiích a otevřených průmyslových standardech. Delmia V5 infrastruktura poskytuje jedinečně škálovatelnou a otevřenou platformu pro společnou výrobu.

Procesní a zdrojové plánovací definice

Používá struktury a schémata řešení procesního plánování v oborech specifických pro danou aplikaci výroby. Ověření metodiky procesu s aktuální produktovou geometrií a definování procesů do podrobnějších detailů.

Řízení výroby

V rámci této sady řešení, jsou zdroje, jako jsou roboty, nářadí, příslušenství, stroje, automatizace a ergonomie, definovány a vkládány do kompletního scénáře výroby.

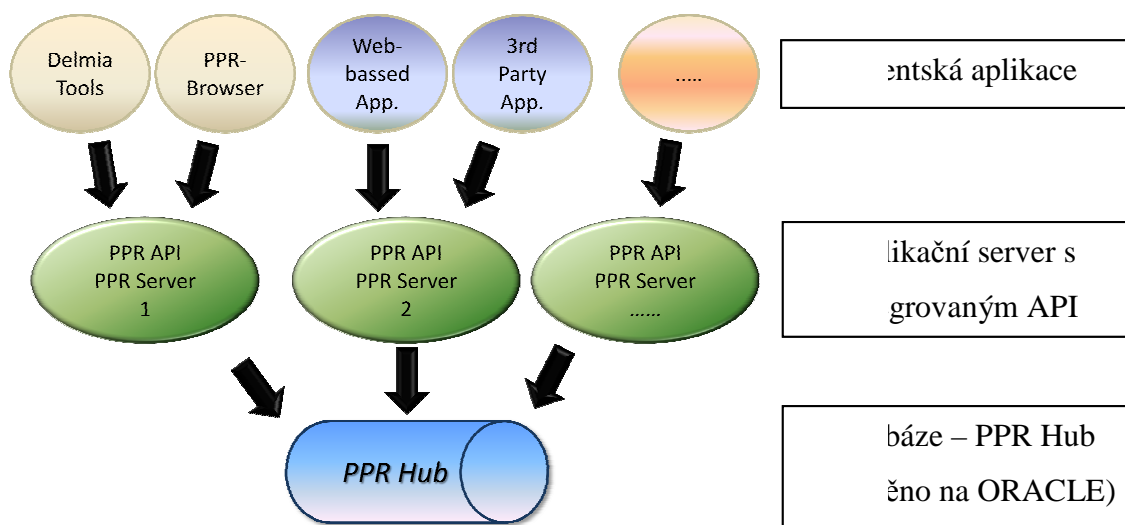
Pro verzi V5 existují tyto moduly:

- **V5 Robotics** – 3D simulace robotických zařízení na pracovišti
- **V5 Ergonomics** – práce s 3D modelem člověka včetně identifikace zatížených částí těla
- **V5 Plant layout** – návrh layoutu linky

Dále existuje část s názvem „Quest“, která se zabývá kompletní simulací celé výroby produktu. Může se použít samostatně nebo jako nástavba DPE. Při použití bez vstupních dat z DPE je nutné veškerá data ručně implementovat do programu.

2.1. DPE z programátorského hlediska

Delmia Process Engineer je databázový systém s třívrstvou architekturou (Klientská aplikace - aplikační server - databáze (Oracle)) (Obr. 2-3).



Obr. 2-3: Schéma DPE

(autor)

Klíčovou komponentou systému je tzv. PPR Hub. Jedná se o objektově relační databázové jádro. Hlavní charakteristikou tohoto jádra je, že umožňuje v rámci použitého datového modelu ukládat každou informaci, která je významná pro integrované produktové a procesní inženýrské činnosti. Každému objektu, který je součástí popisu libovolného výrobního procesu lze pomocí volně konfigurovatelných atributů přiřadit popisné informace. Také je možné definovat nové objekty odvozováním od základních objektů děděním. Pro provádění těchto úprav je v DPE k dispozici konfigurační nástroj. Zde je k dispozici kompletní seznam tříd se všemi atributy a grafickými vlastnostmi. Práce v konfiguračním nástroji nevyžaduje programování v kódu programovacího jazyka. V konfiguračním nástroji nelze přidávat třídám metody. Má-li být součástí customizace také změna chování objektu, je nutné pomocí PPR-API nové funkce naprogramovat.

PPR Hub je díky vlastnostem použitého datového modelu také integračním prvkem mezi jednotlivými aplikacemi z portfolia DS. PPR Hub umožňuje bezproblémovou výměnu dat různých typů.

Všechny objekty, které jsou pro práci v DPE používány jsou odvozeny od jediné abstraktní třídy. Tato třída neumožňuje přímo vytváření objektů, jsou z ní ale odvozeny všechny ostatní třídy, které již mohou být instancovány.

Existují dva základní druhy tříd

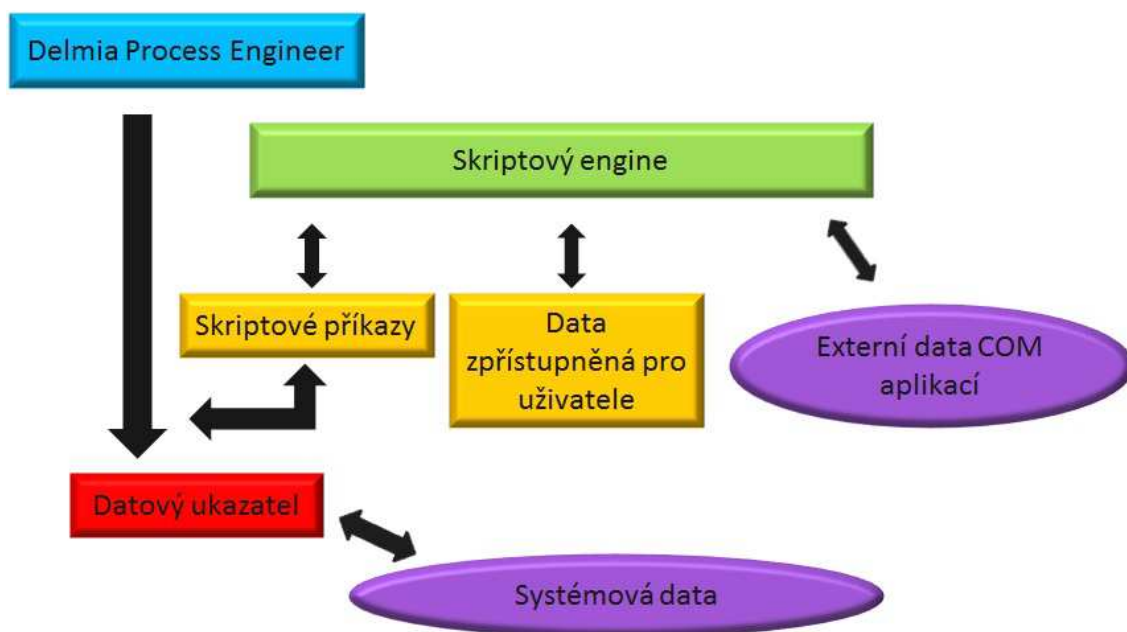
- konfigurační
- plánovacích objektů

Konfigurační třídy jsou součástí základního datového modelu. Možnosti zásahů, bez hlubokých znalostí datového modelu, do těchto tříd jsou velmi omezené. Zásahy do těchto tříd mohou vést k chybnému chování systému. Mezi konfigurační třídy patří například třídy relací mezi objekty, kterých je několik typů.

Třídy plánovacích typů jsou třídy objektů používaných uživateli k vytváření výrobních plánů, představující konkrétní reálné objekty jako je např. součást, nástroj, výrobní operace. Tyto třídy lze volně konfigurovat.

Oba základní druhy tříd jsou ještě rozděleny na tři skupiny, podle toho jestli představují produktový, procesní nebo zdrojový typ objektu. Mezi konfiguračními třídami jsou ale i třídy obecné, z tohoto pohledu nezařaditelné.

Další důležitou komponentou je PPR - API (aplikační programové rozhraní).
(Obr. 2-4)



Obr. 2-4: Schéma PPR - API

(autor)

PPR API umožňuje relativně jednoduchým způsobem programovat jednoduché kontrolní funkce, dávkové úpravy dat, programovat rozhraní pro výměnu dat s externími systémy atd. API nabízí celou škálu tříd s metodami pro přístup k objektům a manipulaci s nimi, a také nástroje pro odchytávání událostí.

Pro spouštění skriptů pod DPE je zapotřebí podpory Windows Script Host, který umožňuje používání skriptů. Funguje pouze pod operačními systémy Microsoft Windows, proto není možné spouštět DPE na jiné platformě. Tento skript host podporuje jazyky: Visual Basic Script, JavaScript, Delphi, PHP a další.

2.2. Customizace DPE

Pro úpravy nebo rozšiřování DPE je zapotřebí ovládat jazyk Visual Basic Script a porozumět integrovanému konfiguračnímu modulu. Důležitou pomůckou je debugger, nutný k ladění běhu programu.

Pro rozsáhlejší úpravy DPE je možné použít implementování vlastní aplikace do celého databázového systému. To znamená, že je možné rozšířit DPE pomocí instalování knihoven vlastních aplikací do prostředí.

2.2.1. Visual Basic Script

Visual Basic Script (dále jen VBS) je skriptovací jazyk vyvinutý společností Microsoft, který vznikl za účelem tvorby webových stránek. Pro jeho používání není potřeba instalovat další software, který by byl potřebný pro programování. V současné době je již každý počítač dostatečně vybaven pro používání tohoto jazyka. Dá se vystačit pouhým textovým editorem a Microsoft Internet Explorerem.

Tento jazyk používá standardní procedury, funkce, cykly – For a While, podmínky - If, různá dialogová okna apod.

```
Dim PlantypeName
sub main(ChildBaseId)
    PlantypeName = Data.GetAttributebyId(ChildBaseId, "plantype_name")
    msgbox PlantypeName
end sub
```

Náhled č. 2-1: Příklad zdrojového kódu ve VBS – vypsání plánovacího typu objektu

Samozřejmostí je i používání komponent jako jsou:

- combo box
- list box
- edit
- check box
- radio button
- date
- FileSelector – otevírání a ukládání souboru.(Open x SaveAs)

Výhody VBS

Pro pokročilého programátora by neměl být problém se s ním naučit pracovat. Je poměrně jednoduché z něj přejít do programování v jazyce Visual Basic a Visual Basic for Applications.

Nevýhody VBS

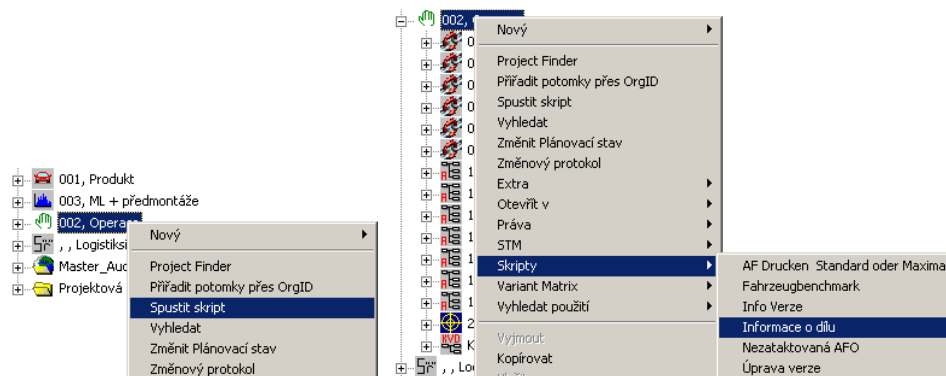
V současné době jde již o poměrně zastaralý jazyk, který se již tolik nepoužívá. Proto by bylo vhodnější používat jiný jazyk, využívající grafické programovací prostředí.

Typy skriptů v DPE a způsoby jejich volání

Pro úpravy v DPE se používají dva typy skriptů. Liší se ve způsobu jejich spouštění. Buď jde o jednorázový skriptový příkaz, nebo o automatickou skriptovou akci.

Skriptový příkaz je jednorázový příkaz, který vykoná určitou operaci s daty. Může být spuštěn ručně uživatelem nebo může být naplánován. Často bývá používán pro hromadné změny atributů operací nebo vazeb operace – díl.

Tento příkaz se spouští na některém nebo konkrétním plánovacím typu ve struktuře (Obr. 2-5). Po této volbě „spustit skript“ se načte seznam všech uložených skriptů, ze kterého je možné jeden vybrat a následně spustit. V knihovně je možné nastavit skriptový příkaz tak, aby ho bylo možné volat z kontextového menu. Takže není nutné při spouštění načítat celou knihovnu skriptů, ale je možné spustit skript pouhým vyvoláním kontextového menu a následným zvolením konkrétního příkazu.



Obr. 2-5: Spouštění skriptového příkazu ze seznamu a z kontextového menu (autor)

Skriptová akce je objekt, který slouží pro odchyťávání událostí na plánovacích objektech. Tomuto objektu je nutné definovat, jaký datový objekt má hlídat, jakou událost má odchytnout, název skriptu, který má spustit a kdy má reagovat. Zda v okamžiku kdy je událost aktivována, tedy před tím než událost proběhne, nebo až po té co skončí. Dále je možno definovat pořadí, akci aktivovat či deaktivovat. (Obr. 2-6) [1]

Obr. 2-6: Okno s nastavováním spouštění skriptové akce

(autor)

Součástí skriptové akce je pak také samotný skript, který musí obsahovat funkci pro odchytnutí dané události. V rámci API je od výrobce softwaru k dispozici konečný počet předdefinovaných deklarací funkcí. Každá funkce umí odchytnout právě jeden typ události (např. smazání objektu, vytvoření vazby mezi objekty, otevření masky vlastností atd.).

Je-li potřeba ošetřit událost, pro kterou neexistuje deklarace funkce od výrobce, je nutné se obrátit přímo na něj.

2.2.2. Visual Basic .NET

Visual Basic .NET je jazyk postavený na platformě „.NET Framework“. Jedná se o moderní objektově orientovaný jazyk, který se neustále vyvíjí a zároveň má velmi širokou základnu vývojářů po celém světě.

VB.NET je poměrně jednoduchý a srozumitelný oproti ostatním jazykům, a to z něj činí jedním z nejvhodnějších jazyků pro začínající programátory. Díky podpoře .NET se z Visual Basicu současně stal plnohodnotný nástroj pro vývoj profesionálních aplikací. A protože se od verze Windows XP stala platforma .NET spolu s jazyky C#, Script.NET a VB.NET nedílnou součástí Windows, je možné VB.NET používat pro tvorbu jednoduchých aplikací namísto skriptovacích prostředí, protože je srovnatelně jednoduchý a současně mnohem výkonnější. Aplikace ve VB.NET běží srovnatelnou rychlostí jako aplikace v C++.

Na rozdíl od skriptovacích jazyků, disponuje VB.NET plnohodnotnou podporou a knihovnamí pro tvorbu grafických aplikací s formuláři, databázovým rozhraním, systémovými funkcemi a rozhraním pro multimédia a hry jako je DirectX nebo OpenGL. [3]

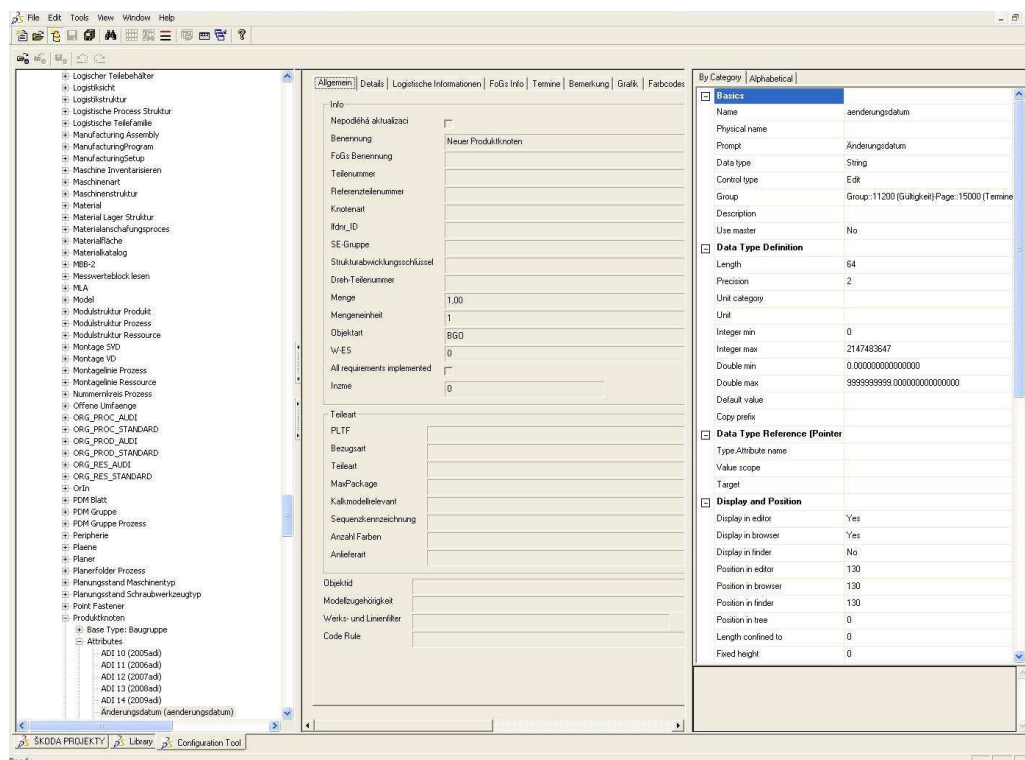
2.2.3. Konfigurační modul

Konfigurační modul je obsáhlý nástroj pro nastavování vlastností objektů používaných v celé aplikaci (Obr. 2-7). Je v něm možné upravovat i chování relací, kterými jsou objekty propojeny

Konfigurační modul je rozdělen na tři okna:

1. Strom sloužící pro procházení plánovacích typů. Zde je znázorněna kompletní struktura objektů, které jsou v celé databázi použity. Od názvu a všech přiřazených atributů až po celý postup vzniku tohoto plánovacího typu, který zahrnuje všechny nadřazené třídy včetně základních abstraktních tříd.
2. Okno pro znázornění aktuálního zobrazení celé masky plánovacího typu. Tato část se používá pro návrh okna vlastností každého objektu. Je zde vidět konečná forma okna, která bude zobrazena při otevírání vlastností na klientovi u uživatele.
3. Seznam vlastností atributů, které je možné editovat. Pomocí tohoto seznamu vlastností atributů, je možné upravovat vlastnosti i chování konkrétních

atributů vybraného objektu. Zároveň je zde možné nastavit jejich viditelnost v masce vlastností.



Obr. 2-7: Okno konfiguračního modulu

(autor)

2.2.4. Propojení DPE s dalšími systémy

Již se základní verzi, kterou dodává výrobce, je možné bez problémů navrhovat kompletní sériovou výrobu. Avšak k dispozici je v tomto případě již upravená verze. Proto je důležité se seznámit se změnami, které byly předem provedeny.

U zákazníka je k dispozici verze DPE s již upravovanou knihovnou plánovacích typů, která je pozměněna kvůli návaznosti dalších systémů používaných ve firmě. Právě tyto systémy se používají v jiných oborech plánování výroby (ekonomické, logistické,...). Kvůli těmto předcházejícím změnám, je nutné důkladně prozkoumat veškeré úpravy, aby nedošlo k poškození procesů provázaných s objekty, do kterých se bude při nových modifikacích zasahovat.

3. Úloha 1

3.1. Zadání

V DPE se vyskytují díly, které se používají jen v některých termínech – např. dle konkrétního (ročního) období. Může se jednat například o kapaliny, ochranné fólie apod. Jsou tedy zařazeny do výroby jen v určitých obdobích.

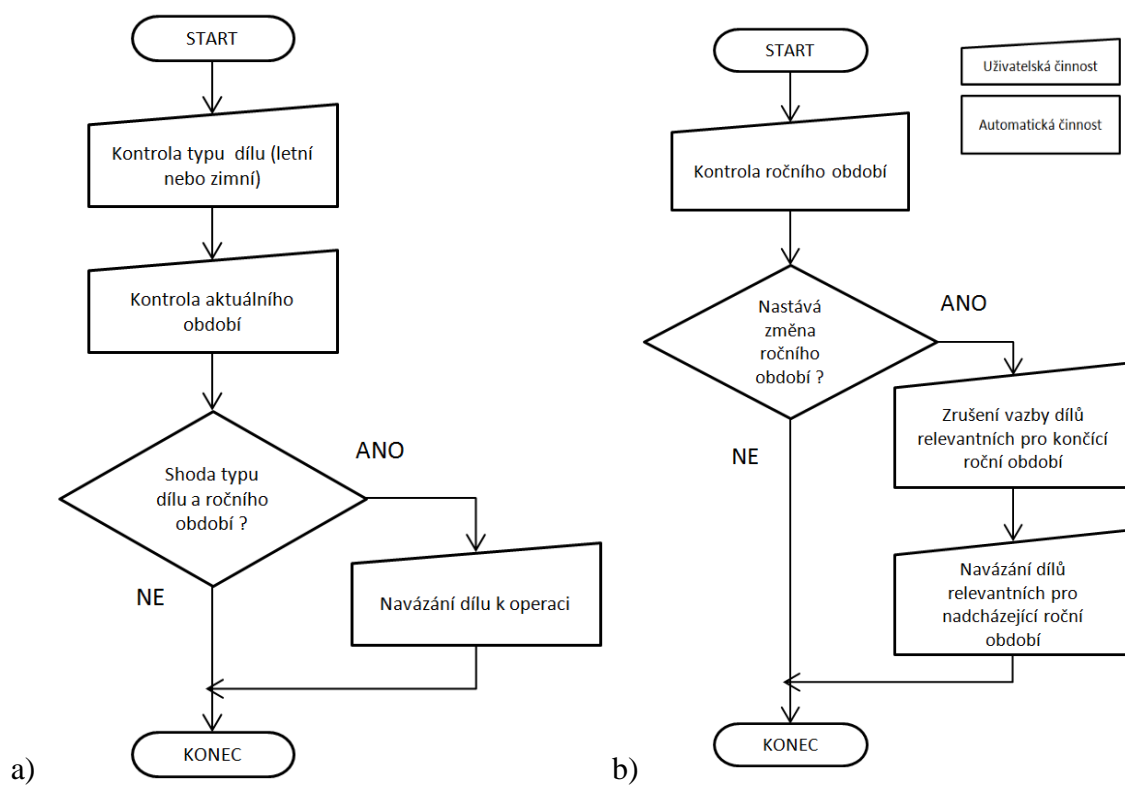
V současné době lze tyto díly v systému používat jen tak, že je musí uživatel před každým obdobím ručně navázat na operaci. Pokud skončí jejich platnost, dojde k odstranění vazby mezi operací a dílem. Tím se díl odstraní z výrobní fáze. Pokud následně začne jeho platné období, musí se opět ručně přiřadit k dané operaci.

Proto si zákazník přál úpravu systém DPE tak, aby bylo možné zadat tyto díly jen jednou a následně se již mohl spolehnout na automatickou úpravu platnosti dílu v každém následujícím období.

3.1.1. Současný stav

Uživatel (technolog) musí provádět tyto činnosti, aby bylo možné vyrábět na výrobní lince produkty, které obsahují díly s odlišnými termíny použití (Obr 3-1). Jde o:

- a) Zavádění nového dílu – tedy o navázání dílu na operaci dle potřebného ročního období. Pro každé období musí technolog navázat správné díly, které se budou používat.
- b) Změnu ročního období. Technolog musí pohlídat, aby se pro každé roční období používal správný díl. To obnáší každého půl roku zkontrolovat a nahradit používané díly.



Obr. 3-1: Schéma současného stavu

(autor)

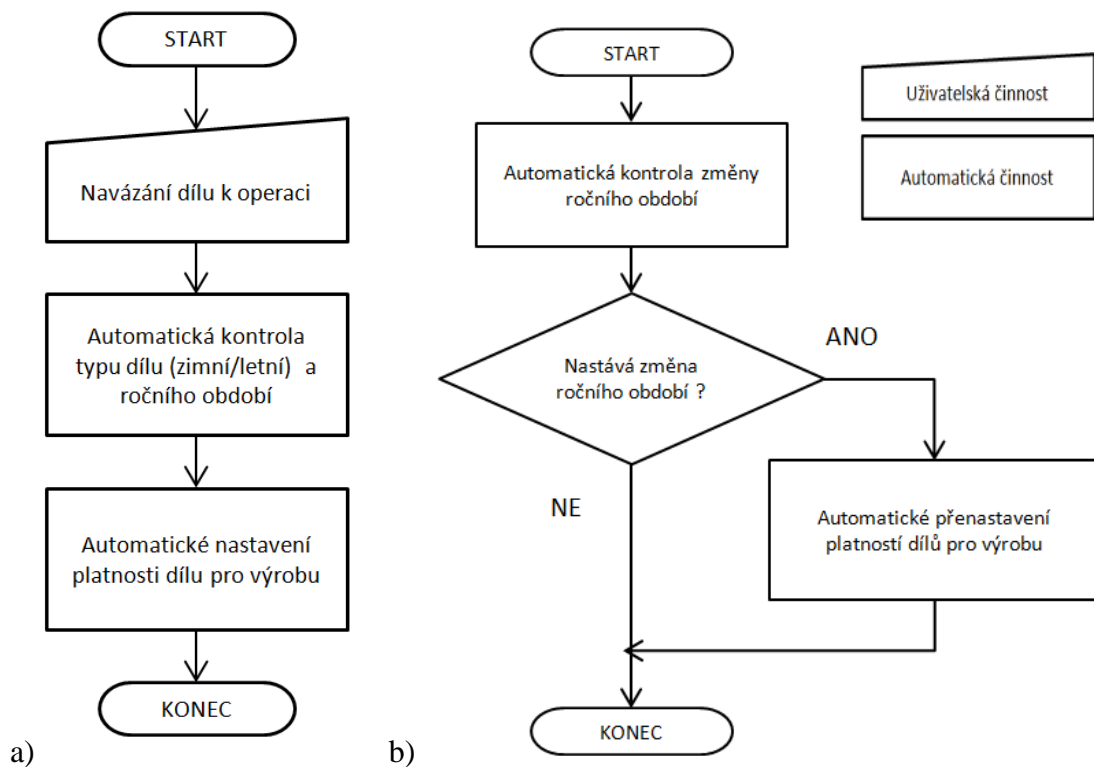
a) Zavádění nového dílu

b) Změna ročního období

3.1.2. Požadovaný stav

Tato úprava má umožnit uživateli zjednodušení práce tak, aby se téměř vše vykonávalo automaticky.

Zavádění nového dílu by obnášelo pouhé provázání na operaci (Obr. 3-2 a). A při změně ročního období by nemusel provádět vůbec žádnou činnost (Obr. 3-2 b).



Obr. 3-2: Schéma požadovaného stavu

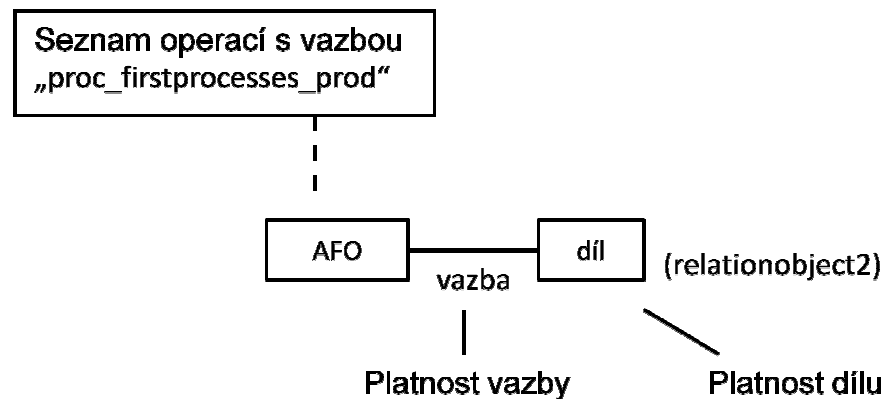
(autor)

a) Zavádění nového dílu

b) Změna ročního období

3.2. Analýza

1. Nejprve je důležité zjistit, která změna má na starosti ovlivnění využívání těchto dílů. Tuto funkci obstarává vazba mezi dílem a operací, na kterou je díl navázán. Vazba, která je pojmenována „proc_firstprocesses_prod“, má jeden z vlastních atributů platnost vazby. Díky této platnosti je díl veden jako používaný (Obr 3-3).



Obr. 3-3: Schéma struktury vazby díl-operace

(autor)

2. Následně upravit všechny stávající díly tak, aby byly využívány podle období. To může být nastaveno spuštěním jednoho skriptového příkazu. Tento skript přiřadí dílům správná období jejich používání tím, že nastaví platnost jejich vazeb na operace se správným termínem platnosti.
3. Po této hromadné úpravě nastavit DPE tak, aby se další díly přiřadily automaticky ke správnému období. Automatické přiřazení může být provedeno spuštěním skriptové akce. Ta provede nastavení platnosti vazby mezi dílem a operací podle toho, ke které operaci se díl přiřazuje.

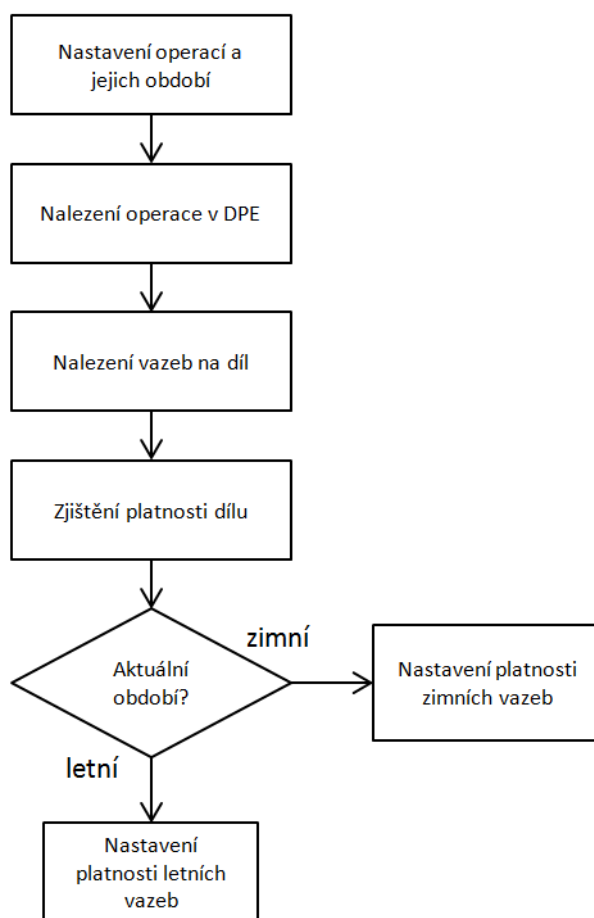
a) Skriptový příkaz

Podle dostupných informací bylo zjištěno, že není možné mít variabilní termín platnosti vazby: díl - operace. Proto tedy bylo nutné navrhnout a napsat skript, který by dokázal pravidelně nastavovat platnosti vazeb dle aktuálního období.

Skript by měl fungovat tímto způsobem (Obr. 3-4):

1. Musí zjistit aktuální datum
2. Dle uživatele nastavit období, která budou vymezovat platnost vazby (díl - operace). Tato období se budou nastavovat na každý následující rok automaticky.
3. Měl by obsahovat seznam operací, které jsou stanoveny uživatelem, pro každá platná období.
4. Následně všechny operace (AFO) ze seznamu najít v celé databázi a u každé zjistit identifikační číslo (ID) vazby typu díl – operace, pokud existuje (Obr. 5-1).
5. Po nalezení těchto vazeb je nutné upravit masku dílu, který je navázán na operaci (pomocí ukazatele: relationobject2).
6. Při nastavování platnosti vazby je nutné zjistit platnost samotného dílu.
7. Dle aktuálního období se pomocí zařazení do příslušného období nastaví požadovaná platnost vazby, která ve výsledku začlení díl do produkce.

8. Pro kontrolu je dobré vytvořit soubor ve formátu „xls“, ve kterém je seznam všech operací a následné nastavení platnosti jejich vazeb s dílem.
9. Jelikož je tento skript připravován jako skriptový příkaz, je nutné nastavit plánované spuštění skriptu před každou požadovanou změnou platného období. To musí provést administrátor příslušného serveru.



Obr. 3-4: Schéma skriptového příkazu

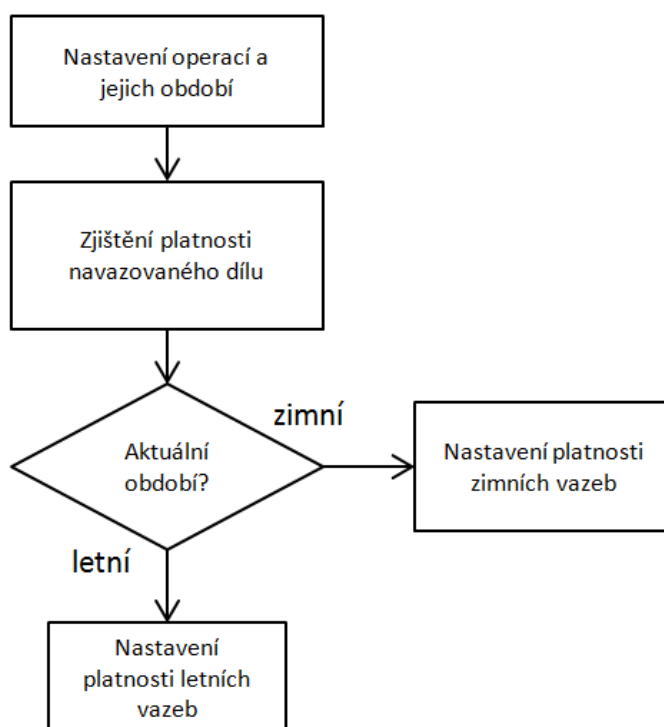
(autor)

- b) Skriptová akce (provádí se při navazování dílu na operaci, která je v seznamu pro různá období)

Je nutné, aby bylo zajištěno automatické nastavování termínu platnosti již při navazování dílu na operaci. Pokud by nebylo toto ošetřeno, mohlo by dojít ke zbytečnému nákupu dílu nebo dokonce k použití tohoto dílu ve špatném období.

Skript tedy musí obsahovat tyto části (Obr. 3-5):

1. Zjištění aktuálního data
2. Vymezení aktuálních platností období – automaticky pro každý následující rok
3. Seznam operací, které chceme ošetřit (letní / zimní)
4. Dle operace, ke které se díl přiřazuje, se nastaví doba platnosti vazby (díl - operace)



Obr. 3-5: Schéma skriptové akce

(autor)

3.3. Řešení

Jelikož jen technolog ví, které díly se mají používat v konkrétních obdobích, není možné tyto díly automaticky rozlišit. Proto je nutné zavést dva typy operací (letní/zimní), do kterých technolog naváže díly podle toho, do kterého období spadají. Po této ruční úpravě je již možné vytvořit tyto skripty.

a) Skriptový příkaz

První krok je tedy zjištění data aktuálního dne. Užitím současného roku se nastaví datum prvního období, měnící platnost vazby (Náhled 3-1).

```
Today = Date  
YearNow = Year (Date)  
DatumS1 = DateSerial(YearNow, 4, 1)
```

Náhled č. 3-1: Ukázka ze skriptu č. 1 – načtení aktuálního data a nastavení prvního období.

Dalším krokem je pevné založení operací, které jsou rozděleny ve dvourozměrném poli na typy zimní a letní.

Po nastavení základních parametrů je možné již přistoupit k prohledávání operací v celé databázi. Pro tuto funkci slouží integrovaná metoda *query*, které je nutné nastavit všechny parametry, podle kterých se má konkrétní plánovací typ najít (Náhled 3-2).

```
Call query.ResetSearch()
Call query.BeginSubQuery("Sub1")
Call query.SetQuery("ergoplantype", "nameshort", "=", "proc")
Call query.SetQuery("ergoplantype", "master", "=", Empty)
Call query.EndSubQuery("Sub1")
Call query.BeginSubQuery("Sub2")
Call query.SetSubQuery("ergoplantype", "master", "=", "Sub1")
Call query.EndSubQuery("Sub2")
Call query.SetSubQuery("ergocompprocessdefault", "plantype", "=", "Sub2")
Call query.SetConcatenator("AND")
Call query.SetQuery("ergocompprocessdefault", "ergoproject", "=", ProjectId)
Call query.SetConcatenator("AND")
```

Náhled č. 3-2: Ukázka ze skriptu č. 1 – query

Následně pomocí metody „Data.GetChildrenCount“ se zjistí počet vazeb, existujících při operaci. Z nich se vybere vazba typu „proc_firstprocesses_prod“.

Po zjištění konkrétního navázaného dílu pomocí metody „Data.GetAttributebyId“, je nutné ověřit platnost dílu a zároveň porovnat s obdobím, ke kterému patří příslušná operace.

Důležité je ošetřit všechny podmínky platnosti vazby a dílu. Může nastat několik situací:

1. Platnost dílu začíná před začátkem nového období a končí po skončení tohoto období

2. Platnost dílu začíná až v průběhu nového období a končí po skončení tohoto období
3. Platnost dílu začíná před začátkem nového období a končí v průběhu tohoto období
4. Platnost dílu začíná až v průběhu nového období a končí v průběhu tohoto období
5. Platnost dílu vůbec nezasahuje do termínu nového období (díl se ještě nepoužívá)

Pro kontrolu je vytvořen soubor ve formátu „xls“, který obsahuje veškeré informace, jako jsou: ID operace, jejich vazby typu díl – operace, navázané díly a jejich platnosti. Na závěr pak konečné termíny platností vazeb, které jsou upraveny dle požadovaného období.

b) Skriptová akce

Ta již využívá některé části použitého kódu ze skriptového příkazu. Seznam operací je vytvořen pomocí metody „Select Case“, kde je k operaci přiřazena hodnota „1“ nebo „0“ podle období, ve kterém se má operace provádět. Zároveň bylo nutné použít již existující funkci, která na základě vytváření vazby spouští konkrétní skript. Tato funkce je definována pod názvem „sa_link“ a má parametry „parent_id“, „children_ids“, „childlistname“ a „children_link_ids“ (Obr. 3-6).

Script entry function <i>function sa_link(parent_id, children_ids, childlistname, children_link_ids)</i>		
Parameters	Description	
	SA_BEFORE	SA_AFTER
children_ids	An array of strings containing the IDs of the (base objects) components to link.	An array of strings containing the IDs of the linked components (link objects).
childlistname	<ul style="list-style-type: none"> up to PE5.13SP6/ PE5.14SP2: A string containing either the name of the plantype, or "nodes", or the name of the type(relationship) which has been created. PE5.13SP7/ PE5.14SP3 and later: A string containing either the GUID of the plantype or the name of the type(relationship) which has been created. 	
children_link_ids	An array of strings containing the IDs of the (link objects) components to link.	

Obr. 3-6: Popis funkce sa_link

(User Manual DPE)

Při použití této funkce bylo nutné nastavit skriptovací akci na hodnotu provádění skriptu „after“, jelikož nebylo možné nastavit hodnotu platnosti vazby ještě před samotným vznikem této vazby. Nejprve tedy bylo nutné vytvořit vazbu a následně bylo možné pracovat s jejími atributy.

Jelikož bylo v DPE vytvořeno datum platnosti dílu v jiném formátu než platnost vazby, bylo zapotřebí vytvořit metodu pro převod tohoto formátu do podoby, kterou vyžadovala vazba. Takže původní formát „yyyymmdd“ bylo nutné převést pomocí jednorozměrného pole na tvar „dd.mm.yyyy“.

3.4. Testování

Pro testování softwaru existují u zákazníka předepsané postupy. Jsou předem vybráni uživatelé, kteří mají za úkol provést všechny operace, u kterých byly provedeny změny nebo kde by mohl nastat nějaký problém. Tito uživatelé musí být předem proškoleni a seznámeni s potřebnými funkcemi k otestování.

Je nutné při testování postupovat dle předem připravených protokolů, kde jsou stanoveny všechny úkony, které je nezbytné provést. Při provádění těchto úkonů se dodržují všechny stanovené postupy a následně se zaznamenávají do protokolu. Pokud se naskytanou problémy, označí se tento krok jako nevyhovující.

Po zavedení těchto úprav DPE proběhlo testování. Při tomto testování bylo nutné nastavit ve skriptu fiktivní datum, na které pak reagovalo nastavení platnosti vazby, která pak určovala použití dílu při montáži. Toto testování přineslo pozitivní závěr, který zněl, že je **vše v pořádku** a je možné tyto úpravy nasadit i na produkční server.

Příklad testování: Pokud bude platnost operace od 1. 4. do 31. 10. a díl bude po celou dobu tohoto období platný (Obr. 3-7), dojde k automatickému nastavení hodnoty na vazbě (díl-operace) dle příslušného období (Obr. 3-8).

Produktový uzel <K, 5J0853405H, FOLIE OCHRANNA, LT, 502931934, >

Obecné | Detaily | Logistické informace | FoGs info | Termíny | Poznámka | Grafika | Barevné kódy | Procesní informace dílu | SKD

Platnost

Konadatum žádané	
KZ Stav vývoje	B
Ukončení_A	99991231
Nasazení_A	20090525
Datum náběhu nulové série	20090525
Vysazení nulové série	99991231

Obr. 3-7: Znáznornění platnosti dílu

(autor)

Produktový uzel <K, 5J0853405H, FOLIE OCHRANNA, LT, 502931934, >

Obecné | Detaily | Logistické informace | FoGs info | Termíny | Poznámka | Grafika | Barevné kódy | Procesní informace dílu | SKD

Ruční úpravy

Taktungsname	New data object
Óznačení sekvence	
Datum náběhu	<input checked="" type="checkbox"/> 1.4.2012
Datum ukončení	<input checked="" type="checkbox"/> 31.10.2012
Množství ručně	0,00
Strana materiálové plochy	

Obr. 3-8: Znáznornění platnosti vazby (díl-operace)

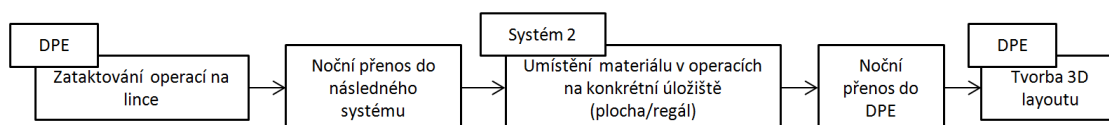
(autor)

4. Úloha 2

4.1. Zadání

4.1.1. Obecný popis

U zákazníka se ve fázi sériového plánování (změny výrobního plánu, sériové změny produktu) používá samostatná aplikace pro přiřazování přepravek, ve kterých jsou používané díly ukládány do konkrétních regálů na úsecích montážní linky. Tato aplikace využívá pouze tabulkového zobrazení umístění materiálu na výrobní lince. Oproti DPE má velmi dobře propracovaný způsob správy úložišť, nabízí více funkcí a je uživatelsky komfortní. Datová výměna s touto aplikací však probíhá dle Obr. 4-1.



Obr. 4-1: Schéma propojení původní aplikace s DPE

Takové provázání není pro plánování nejvhodnější. Data nejsou v aplikaci aktuální, a tudíž může nastat spousta potíží. Uživatel nemůže zařadit nové díly do správného úložiště, protože nový díl není v tu stejnou dobu zaveden v aplikaci. Zároveň, naopak po zrušení používání dílu, se díl v aplikaci ještě vyskytuje.

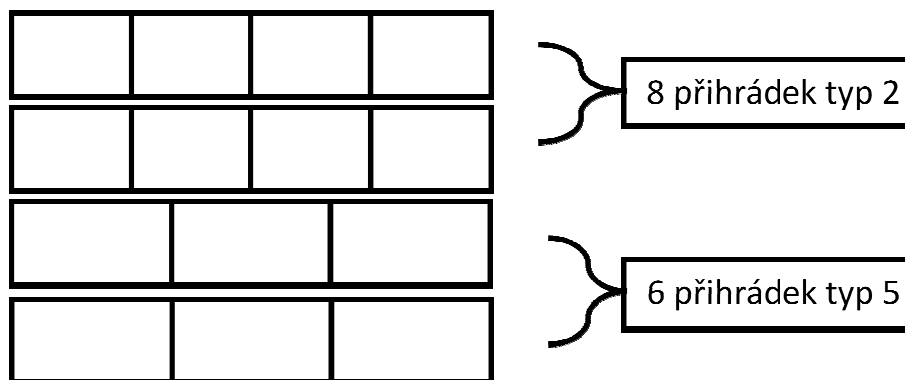
Kvůli takovému propojení je třeba vytvořit novou funkci do DPE, která umožní práci s nejaktuálnějšími daty a zároveň bude schopna generovat strukturu úložišť.

Zákazník měl tedy požadavek: „zimplementovat do systému DPE přiřazování přepravek k úložištím“. K tomuto úkolu samozřejmě patří rozsáhlá příprava dat. Dále pak přehledné a jednoduše ovladatelné uživatelské prostředí, které bude ošetřovat veškeré potřebné kroky při přiřazování dílů s přepravkami do jednotlivých regálů. Po těchto úpravách je zapotřebí převést všechna data zpět do databáze a opět je zpracovat do struktury.

4.1.2. Používaná úložiště

Zákazník používá standardizované typy regálů o konstantních vnějších rozměrech. Vnitřní uspořádání přihrádek v regálu je variabilní a je dáno používanými přepravkami.

Konkrétní typ regálu je definován počtem přihrádek pro typ přepravky v jednotlivých patrech (Obr. 4-2).



Obr. 4-2: Standardní regál 0-8-0-0-6

(autor)

Zákazník používá pět typů regálových přepravek (Tabulka č. 1). Stejně používá pět typů přihrádek (Tabulka č. 2). Přihrádky jsou kolejnicové bez pevného dna.

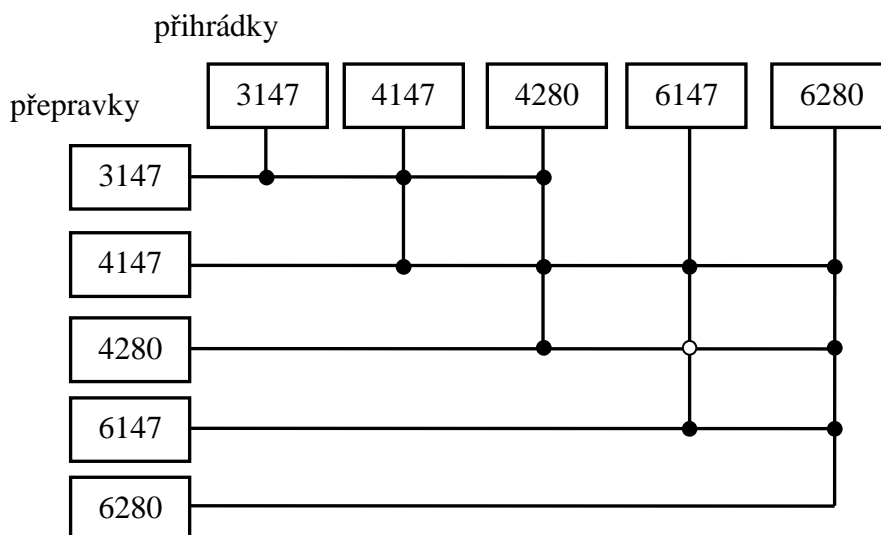
Typ přepravky	šířka	výška	hloubka
3147	198	140	297
4147	297	140	396
4280	297	280	396
6147	396	140	594
6280	396	280	594

Tabulka č. 1: Rozměry používaných přepravek

Typ přihrádky	šířka	výška
3147	200	140
4147	300	140
4280	300	280
6147	400	140
6280	400	280

Tabulka č. 2: Rozměry používaných přihrádek v regálu

Z tabulek vyplývá, že lze menší přepravky umístit i do větších přihrádek otočením o 90°. Popřípadě nižší přepravku lze umístit do vyšší přihrádky. Pro tuto možnost je navržen diagram, který znázorňuje v řádku přihrádky a ve sloupci přepravky. Dle propojení je možné vyčíst, jakou přepravku je možné vložit do které přihrádky (Obr. 4-3).

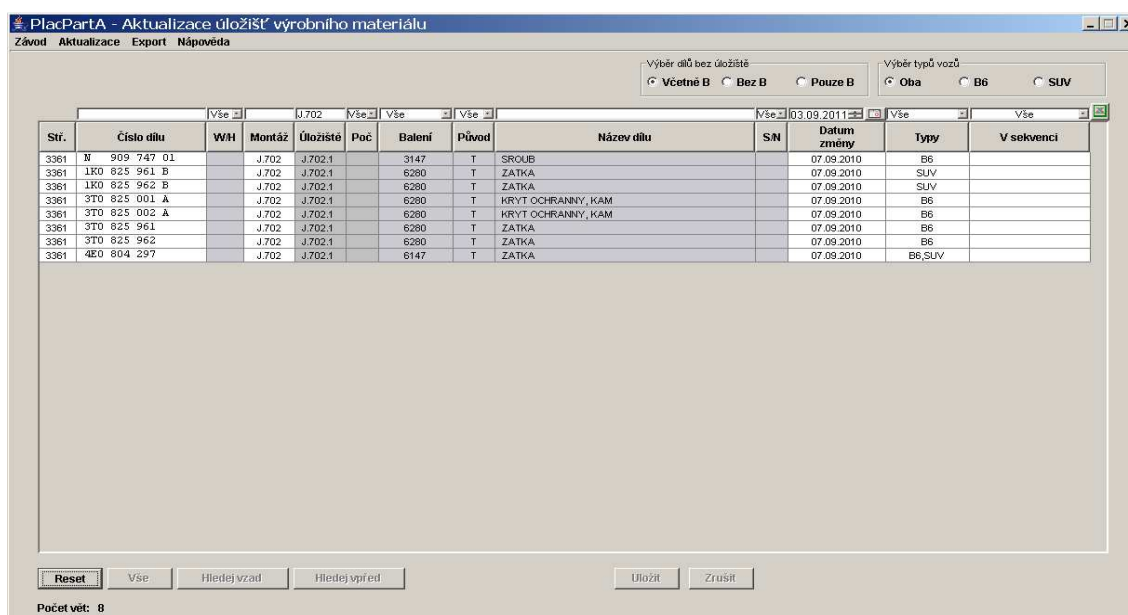


Obr. 4-3: Diagram možností uložení přepravek v regálu

(autor)

4.1.3. Požadavky na uživatelské prostředí

Zákazník požaduje přiblížit uživatelské prostředí původní aplikaci (Obr. 4-4).



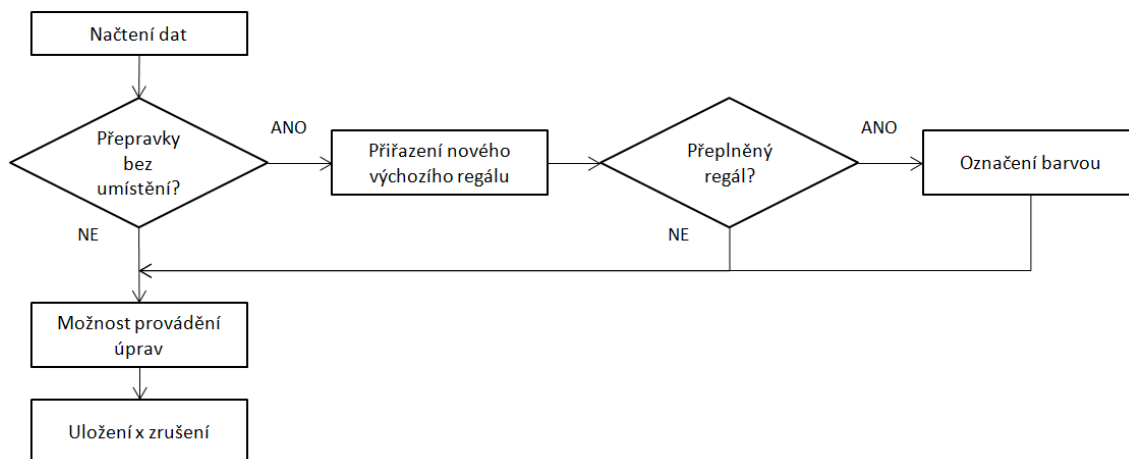
Obr. 4-4: Vzhled původní aplikace

(autor)

Pracovní plocha u uživatele má mít podobu interaktivní tabulky, která zobrazí všechny relevantní údaje. Dále má být součástí plochy lišta s jednotlivými menu pro volání funkcí.

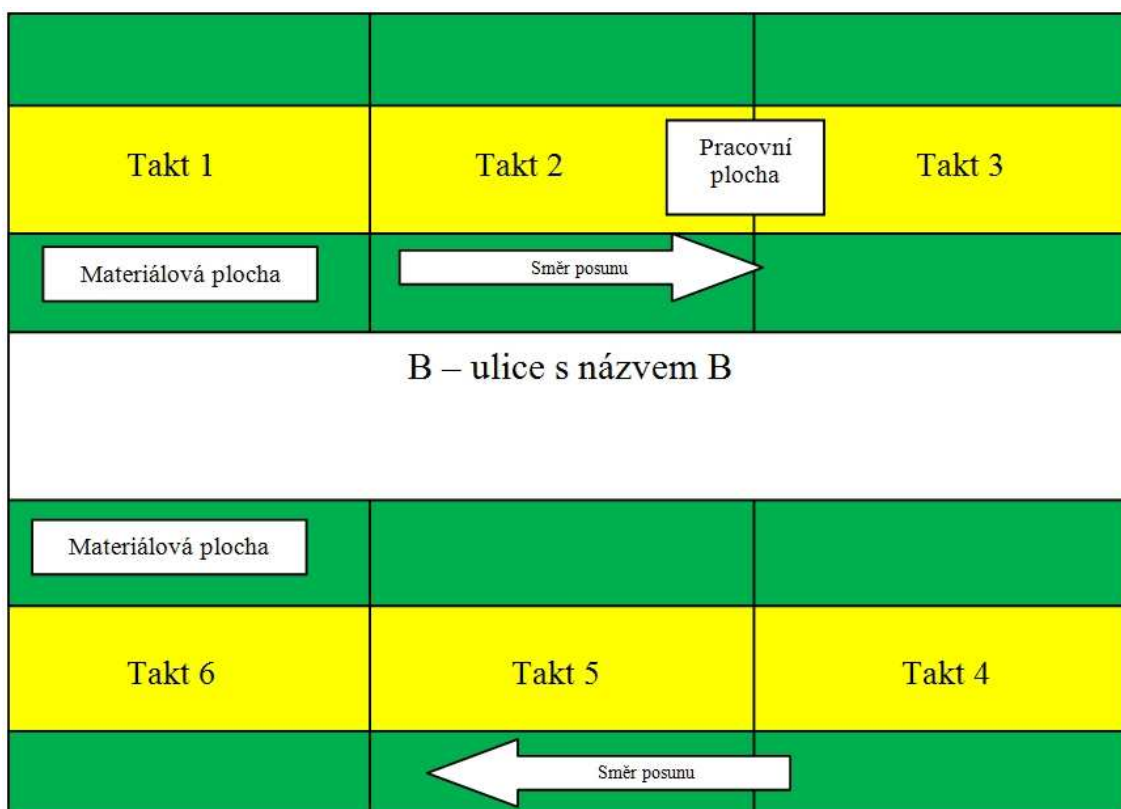
4.1.4. Požadované funkcionality a chování aplikace

Aplikace by měla splňovat tyto základní funkce (Obr. 4-5).



Obr. 4-5: Schéma požadovaných funkcí

1. Při spuštění dojde k načtení dat z DPE
2. K dílům s dosud nepřirazeným regálem se přiřadí nový regál, který bude mít v názvu kompletní „adresu“ umístění, která bude obsahovat: ulici umístění, označení taktu a číslo regálu (Obr. 4-6). Zároveň bude označen jinou barvou, kvůli rozpoznání od již existujících regálů.
3. Tento název regálu bude mít možnost uživatel ručně přepsat, a tím založit nový regál dle potřeby rozmístění dílu v taktu.
4. Při generování nových regálů dojde k jejich naplnění přepravkami. Pokud bude regál přeplněn, bude označen - zvýrazněn odlišnou barvou.
Logika stávající používané aplikace rozpozná, v případě umístění přepravky do regálu, zda v případě zaplnění typově shodných přihrádek, je volná použitelná větší přihrádka a přepravku tam umístí. Nemůže-li přepravku umístit, ohlásí přeplněnost.
5. Po uspořádání všech dílů do regálů, by mělo být možné tyto změny uložit nebo je úplně zrušit.
6. Při ukládání by se měly tyto změny ihned projevit ve struktuře DPE.



Obr. 4-6: Náskres linky

(autor)

4.2. Analýza

4.2.1. Příprava a načtení dat

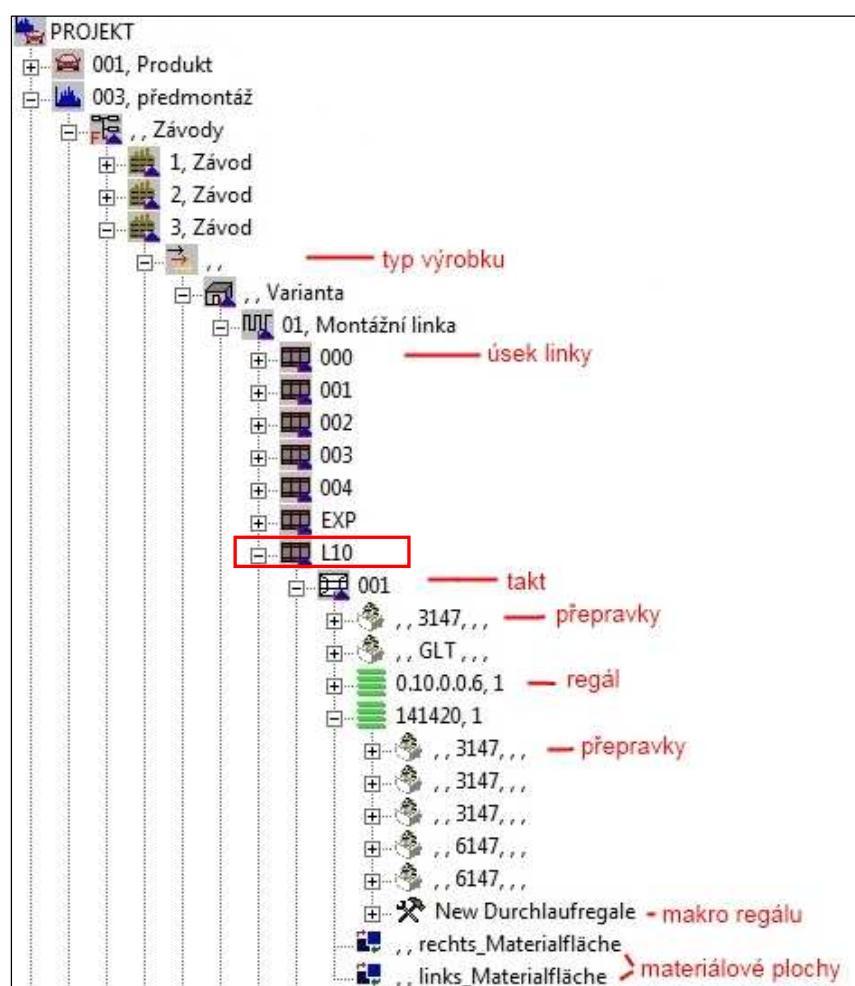
Taktovací struktura je vyobrazena stromem, obsahujícím závod, typ výrobku, variantu montáže, konkrétní linku a její úsek. A nakonec určitý takt neboli montážní stanoviště (Obr. 4-7).

Je nutné v taktovací struktuře založit nový úsek, který bude obsahovat pouze data pro tvorbu layoutu. Obvykle se zde vyskytuje úsek, obsahující informace o taktech, ke kterým jsou přiřazeny operace s navázanými díly. Někdy i konkrétní dělník, který tyto operace vykonává. Je tedy potřeba založit nový uzel LAYOUT, obsahující pouze informace o taktu, jeho regálech, a přepravkách v něm obsažených (Obr. 4-8). Této části se bude říkat struktura úložišť.



Obr. 4-7: Struktura výrobní linky

(autor)



Obr. 4-8: Struktura výrobní linky s novým uzlem

(autor)

Všechny informace o balení je nutné aktualizovat pomocí spuštění jediného skriptového příkazu. Tento skript by měl porovnat aktuální data s naposledy generovanými, a připravit dávku dat pro následnou uživatelskou úpravu.

4.2.2. Nová implementovaná aplikace - vzhled

Uživatel by měl mít k dispozici přehledné okno, zobrazující všechna potřebná data, obsahující informace o dílech na konkrétním úseku montážní linky. Tyto díly zde budou mít přiřazen kód typu přepravky a konkrétní označení regálu, do kterého budou na montážní lince vkládány.

Pro jednoduchou práci s vyhledáváním, by zde měli být k dispozici různé typy filtrů. Pro vyhledávání v textu je možné použít jednoduchý textbox. Pro odfiltrování atributů s omezeným počtem typů je možné použít combobox, který bude automaticky naplněn dle vyskytujících se hodnot v tabulce. U atributu obsahujícího datum je možné použít textbox, propojený s monthcalendar, nebo samotný datetimestpicker.

4.2.3. Nová implementovaná aplikace – dialogová okna

Pro usnadnění uživatelské práce bude třeba vytvořit několik dalších dialogových oken:

- Hlavní okno
- Nastavení typové přepravky
- Umístění dílu s přepravkou do regálu
- Změna úložiště
- Varovná hlášení

4.2.4. Návrat do DPE

Při prvním spuštění se po uložení změn vygeneruje nový úsek struktury úložišť. Pokud se již někdy na tomto úseku pracovalo, byla pak vygenerována struktura, která bude nyní přepsána stávajícími daty. Zde budou generovány všechny regály včetně jejich vazeb s přepravkami konkrétních dílů.

4.2.5. Převravy

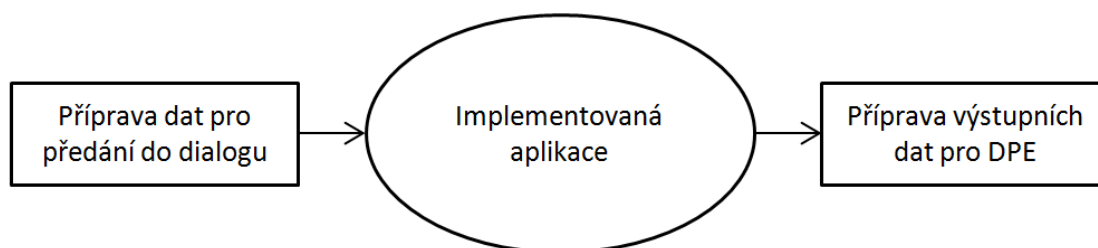
Všechny montované díly jsou ukládány do přepravek o různých rozměrech. Proto se používá na výrobní lince 5 nejběžnějších druhů přepravek (KLT), které se budou v regálech vyskytovat. Jelikož mají tyto přepravy různé rozměry, je nutné pro ně v regálech vytvořit 5 různých přihrádek. Pro některé přepravy menších rozměrů, je možné využít volných přihrádek, určených pro větší přepravy (Obr. 4-3).

Všechny tyto regály mají tedy určené, kolik a kterých přepravek s používanými díly je možné uložit do konkrétního regálu.

Zároveň se používají ještě velké přepravy (GLT), které se nekládají do regálu, ale mají místo pouze na montážní ploše (např. vedle regálu).

4.3. Řešení

Tuto úlohu je třeba řešit pomocí skriptových příkazů pro přípravu databázových dat, naprogramováním nové aplikace s následným vygenerováním knihovny dll, která bude implementována do DPE. A v neposlední řadě dalším skriptem pro převzetí upravených dat zpět do DPE a vygenerování nové struktury úložišť (Obr. 4-9).



Obr. 4-9: Tři hlavní části úlohy

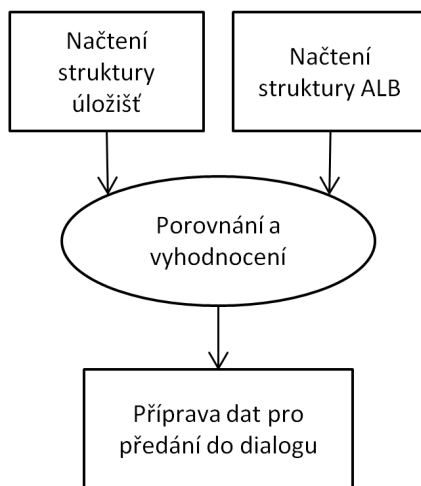
(autor)

4.3.1. Skripty upravující data pro implementovanou aplikaci

Pro tuto část úlohy slouží 2 skripty. „MaterialPlanning“ a „MaterialPlanning_engine“.

První zmiňovaný obsahuje funkce na otestování existence knihovny v systému, spouštění funkcí z druhého skriptu, který pomáhá zjistit rozdíly taktovacích struktur v DPE. Tento skript je, po zavedení do DPE, spouštěn jako skriptový příkaz.

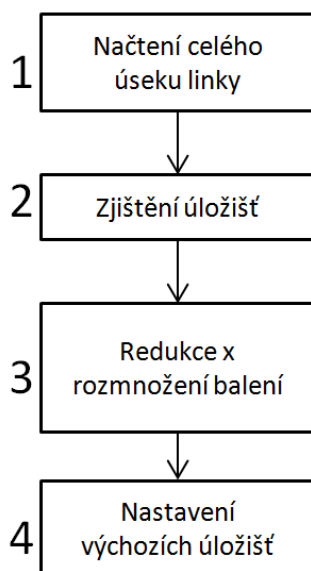
MaterialPlanning_engine je skript, který načítá strukturu úložišť (pokud již existuje) a ALB taktovací struktury. Tyto dvě struktury porovnává, a zjišťuje veškeré změny v uložení dílů do regálů (Obr. 4-10).



Obr. 4-10: Funkce skriptů pro přípravu dat

(autor)

Načítání struktury úložišť probíhá tímto způsobem (Obr 4-11):



Obr. 4-11: Schéma načítání struktury úložišť

1. Načte data všech taktů na konkrétním uzlu úseku, na kterém se skriptovací příkaz spouští.

2. Zjistí takt, operaci (AFO), ulici umístění, číslo dílu, balení dílu (pokud existuje) a další informace.
3. Balení dílů v rámci jednoho taktu:
 - Zredukuje do jednoho balení: pokud je více dílů se stejným číslem
 - Rozmnoží balení - vytvoří se nové (kopie): pokud se díl používá ve více variantách (barvách)
4. Úložiště balení nastaví na výchozí hodnotu
 - Pro malé přepravky (KLT) → Ulice.Takt.1
 - Pro velké přepravky (GLT) → Ulice.Takt

Při načítání a porovnávání změn se mohou vyskytnout tyto možnosti:

- a) Díl zůstává na původním místě.
- b) Díl se přesunul.
- c) Byl použit nový díl.
- d) Díl, který byl dříve použit, již není zařazen.
- e) Díl byl přebalen do jiného druhu přepravky.

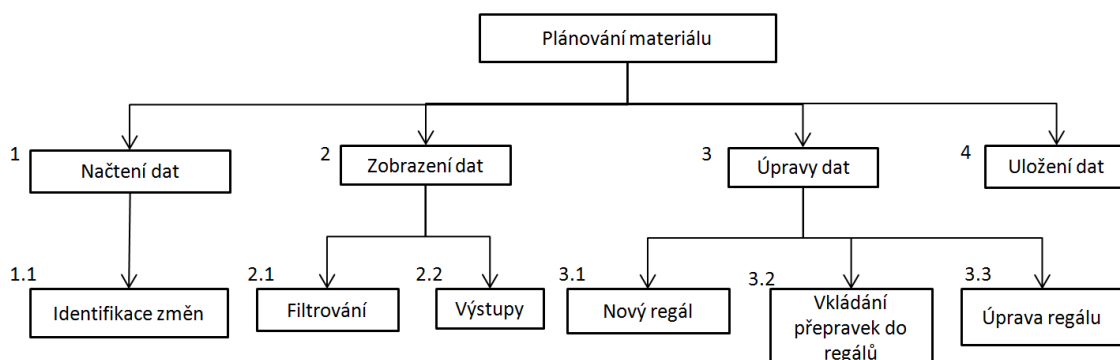
Následně se zavolá metoda pro načtení dat a zobrazí se tzv. hlavní dialogové okno. Skript předá funkci knihovně a čeká na návrat.

4.3.2. Implementovaná aplikace

Tato aplikace je vytvářena v prostředí MS Visual Studio 2010 v jazyce Visual Basic.NET, který je vybrán kvůli podobnosti s jazykem VB Script a jeho možnostem upravování ostatními vývojáři spravující systém DPE.

Tato knihovna je implementována do systému. Přebírá od skriptu všechna potřebná data z DPE, která se zde upraví a následně předává zpět skriptu.

Aplikace má na starosti (Obr. 4-12):



Obr. 4-12: Diagram hlavních funkcí aplikace

(autor)

- 1 → Přebírání dat z DPE od skriptu.
- 1.1 → Identifikace změn od předchozího stavu
- 2 → Souhrnné a přehledné zobrazení dat.
- 2.1 → Filtrování dat pro jednodušší vyhledávání.
- 2.2 → Umožnění exportu
- 3 → Úpravy dat
- 3.1 → Zakládání nových regálů
- 3.2 → Ukládání přepravek do regálů
- 3.3 → Změna umístění přepravky (i hromadná)
- 4 → Uložení a předání upravených dat

Aplikace obsahuje tyto dialogy:

- Hlavní formulář obsahující zobrazující tabulku, filtry a tlačítka.
- Dialog pro vytvoření nového regálu, který je vyvolán z menu, nebo zadáním zatím neexistující adresy ve sloupci úložiště. Dále je použit při výběru existujícího regálu pro přepravku.
- Dialog zobrazující návrh kapacity regálu a jeho aktuální obsazení přepravkami. Je zobrazován při vytváření nového regálu nebo při úpravě již existujícího regálu.

- Dialog pro změna typu přepravy pro číslo přepravy
 - přiřazení typu k neznámému číslu přepravy
 - změna typu přepravy pro známou přepravku

Aplikace obsahuje tyto třídy:

- Třída pro návrh plnění regálu
 - zjišťuje přeplnění regálu
 - volnou kapacitu regálu
- Třída pro práci s daty Layoutu
 - seznam středisek, jejich výchozích regálů, zavedených ulic a přiřazených neznámých přepravek
 - zjištění existence úložiště (Ulice.Takt /pro GLT/ nebo Ulice.Takt.CisloRegalu /pro KLT/)
 - vytvoření nového úložiště
- Třída osahující různá aplikační nastavení (oddělovače, výchozí hodnoty, cesta k helpu)

Filtrování

Jednou z hlavních funkcí této aplikace je filtrování. Ta zjednodušuje uživateli práci výběrem používaných dat.

Pro zadávání podmínek výběru dat jsou zde použity komponenty: textbox, combobox a datetimerpicker.

Textbox funguje filtrování tak, že se do okna zadá alespoň část názvu hledaného atributu a potvrdí se pomocí klávesy „enter“.

Combobox je naplněn všemi hodnotami, které se v aktuálním sloupci vyskytují. Dále jsou tyto hodnoty, kvůli přehlednosti, srovnány.

Datetimerpicker omezuje výběr podle toho, které datum v kalendáři se nastaví. Pokud je datum nastaveno, zobrazí se odfiltrované údaje, které byly změněny od příslušného data.

Ve zdrojovém kódu jsou metody pro filtrování, plnění comboboxů hodnotami z příslušných sloupců, a samozřejmě metoda pro vyprázdnění filtrujících komponent a zobrazení všech načtených dat.

Pro přehlednost použitých filtrů je použito obarvování filtrů. To je provedeno tak, že pokud je filtr aktivní, zvýrazní se konkrétní komponenta filtru zelenou barvou.

Počty záznamů

Pro uživatele je zde vytvořen přehled obsažených typů záznamů. V něm je hned vidět, kolik a jakých změn nastalo od posledního spouštění tohoto materiálového plánování.

Ošetření tlačítek

Před ukončením materiálového plánování je nutné všechny změny potvrdit uložením. Tím se zapíše všechny změny do DataTable a následně je možné tuto aplikaci ukončit. Pokud by ji chtěl uživatel ukončit bez uložení, je následně upozorněn na možnost zrušení všech předchozích změn.

Ukončení

Funkce uložit zapíše změny jen v datech v paměti DataTable. Výsledné ukládání a tvorba nové struktury v DPE je spouštěna až při uzavření okna materiálového plánování. Tím se ukončí zápis do DataTable a následně převezme všechny funkce skript pro tvorbu dat v DPE.

Primární okno aplikace

Základní okno nové aplikace (Obr. 4-13) je navrženo tak, aby splňovalo podmínky jednoduchosti a přehlednosti pro technology a průmyslové inženýry, kteří s ní budou pracovat.

Aktualizace úložíšť výrobního materiálu

Střediska Akce Návoděda

Reset Vše Vše Vše Vše Vše 08.03.2012 Vše Vše Vše

Str.	Číslo dílu	W/H	Montáž	Úložiště	Poč.	Balení	Původ	Název dílu	S/N	Datum změny	Typy	Oper.	Sekv.
3361	370 837 085 A	J	G.200	G.200		111902 (7)	K	BOWDEN	N		B6	3018	
3361	370 837 085 B	XXX	J	G.200	G.200	522219 (7)	H	TRMEN LOŽISKA	N		B6	3017	
3361	370 839 085 C	J	G.200	G.200		111902 (7)	K	BOWDEN	N		B6	3017	
3361	379 839 702	J	G.200	G.200		114888 (7)	K	TESNENÍ DVERÍ VNEJ	N		B6	3024	
3361	510 837 085 A	J	G.200	G.200		PLKT_6 (7)	K	BOWDEN	N		SUV	3018	
3361	510 837 702 A	J	G.200	G.200		114888 (7)	K	TESNENÍ DVERÍ VNEJ	N		SUV	3024	
3361	510 837 085 C	XXX	J	G.200	G.200	522219 (7)	H	TRMEN LOŽISKA	N		SUV	3018	
3361	510 839 085 A	J	G.200	G.200		PLKT_6 (7)	K	BOWDEN	N		SUV	3017	
3361	510 839 702	J	G.200	G.200		114888 (7)	K	TESNENÍ DVERÍ VNEJ	N		SUV	3024	
3361	510 839 085 A	XXX	J	G.200	G.200	522219 (7)	H	TRMEN LOŽISKA	N		SUV	3017	
3361	510 839 085 H	XXX	J	G.200	G.200	522219 (7)	H	TRMEN LOŽISKA	N		B6	3017	
3361	M72B	S	G.200	G.200		005280 (7)	S	ZAMEK DVERÍ	N		B6,SUV	3017,3018	M72B
3361	S17B	S	G.200	G.200		x (7)	S	SKLO DVERÍ ZADNÍCH	N		B6,SUV	3451	
3361	MHT 003 849	J	G.200	G.200.1		4147	K	CEP	N		B6,SUV	3451	
3361	370 837 085 A	J	H.200	H.200		111902 (7)	K	BOWDEN	N		B6	3016	
3361	370 839 085 C	J	H.200	H.200		111902 (7)	K	BOWDEN	N		B6	3015	
3361	379 839 701	J	H.200	H.200		114888 (7)	K	TESNENÍ DVERÍ VNEJ	N		B6	3023	
3361	510 837 085 A	J	H.200	H.200		PLKT_6 (7)	K	BOWDEN	N		SUV	3016	
3361	510 837 701	J	H.200	H.200		114888 (7)	K	TESNENÍ DVERÍ VNEJ	N		SUV	3023	
3361	510 837 085 B	XXX	J	H.200	H.200	522219 (7)	H	TRMEN LOŽISKA	N		SUV	3016	
3361	510 839 085 A	J	H.200	H.200		PLKT_6 (7)	K	BOWDEN	N		SUV	3015	
3361	510 839 701	J	H.200	H.200		114888 (7)	K	TESNENÍ DVERÍ VNEJ	N		SUV	3023	
3361	510 839 085 A	XXX	J	H.200	H.200	522219 (7)	H	TRMEN LOŽISKA	N		SUV	3015	
3361	510 839 085 H	XXX	J	H.200	H.200	522219 (7)	H	TRMEN LOŽISKA	N		B6	3016	

Počet záznamů: 150 150 nové 0 storn. 0 změny

Uložit Zavřít

Obr. 4-13: Hlavní formulář implementované aplikace (autor)

Obsahuje základní komponentu datagridview, využívanou pro zobrazování dat v tabulkové podobě. Nad komponentou datagridview jsou další použité komponenty jako např. textbox, combobox a datetimepicker, které jsou využity pro filtrování dat zobrazených v tabulce. Toto filtrování má usnadnit uživateli jednoduchý přístup k hledaným datům pomocí zobrazení dílů splňujících podmínky filtrování.

Ve vygenerované tabulce je možné měnit úložiště dílu (regál na pracovišti). To je umožněno jednoduchým kliknutím levého tlačítka myši na konkrétní úložiště a následným nastavením hodnoty dle požadovaného regálu.

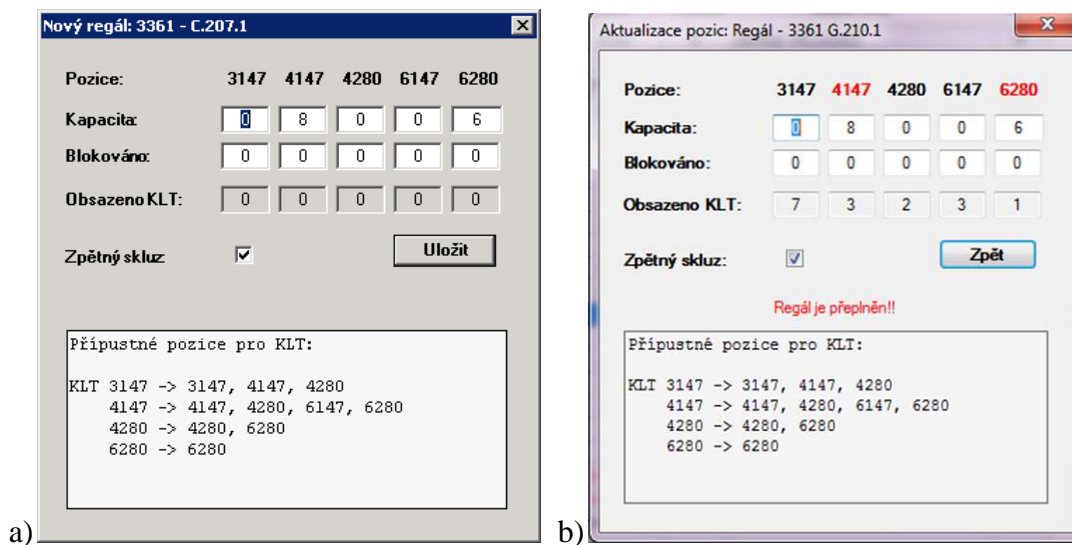
Po přiřazení dílu k regálu, dojde k vyhodnocení obsazenosti regálu. Pokud je již přeplněn, je o tom uživatel informován hlášením a zároveň se zvýrazní jeho název červenou barvou. Pokud má regál stále volné pozice, zůstane bez změny barvy.

Pro zjednodušení a urychlení práce, je zde aktivní funkce pro zobrazování základních údajů. Ta umožní, při najetí myši na konkrétní políčko tabulky, zobrazit některé vlastnosti vybraného objektu.

Okno pro přiřazování přepravek do regálů

Okno pro přiřazování přepravek do regálů je navrženo tak, aby jednoduše popsalo typ regálu s počtem různých typů přihrádek a obsazenost těchto jednotlivých přihrádek.

První řádek znázorňuje počet konkrétních přihrádek, které regál obsahuje. Druhý řádek je připraven pro aktuální blokování přihrádek pro následné vedlejší účely. Třetí řádek ukazuje, kolik je již obsazených přihrádek v regálu před těmito úpravami. Toto dialogové okno je navrženo tak, aby uživatel mohl jednoduše obsazovat volné přihrádky a nemusel zjišťovat, zda se konkrétní potřebná přepravka do regálu ještě vejde. Diagram (Obr. 4-3), který popisuje možnosti uložení přepravek do přihrádek, bylo nutné převést do algoritmu, který hlídá počet možných volných pozic v regálu. Pokud dojde k přeplnění regálu, je s tím uživatel ihned seznámen a zároveň se mu vyznačí přihrádky, které musí upravit, aby bylo splněno správné plnění regálu (Obr. 4-14).



Obr. 4-14: Náhled na okno s ukládáním KLT- přepravek

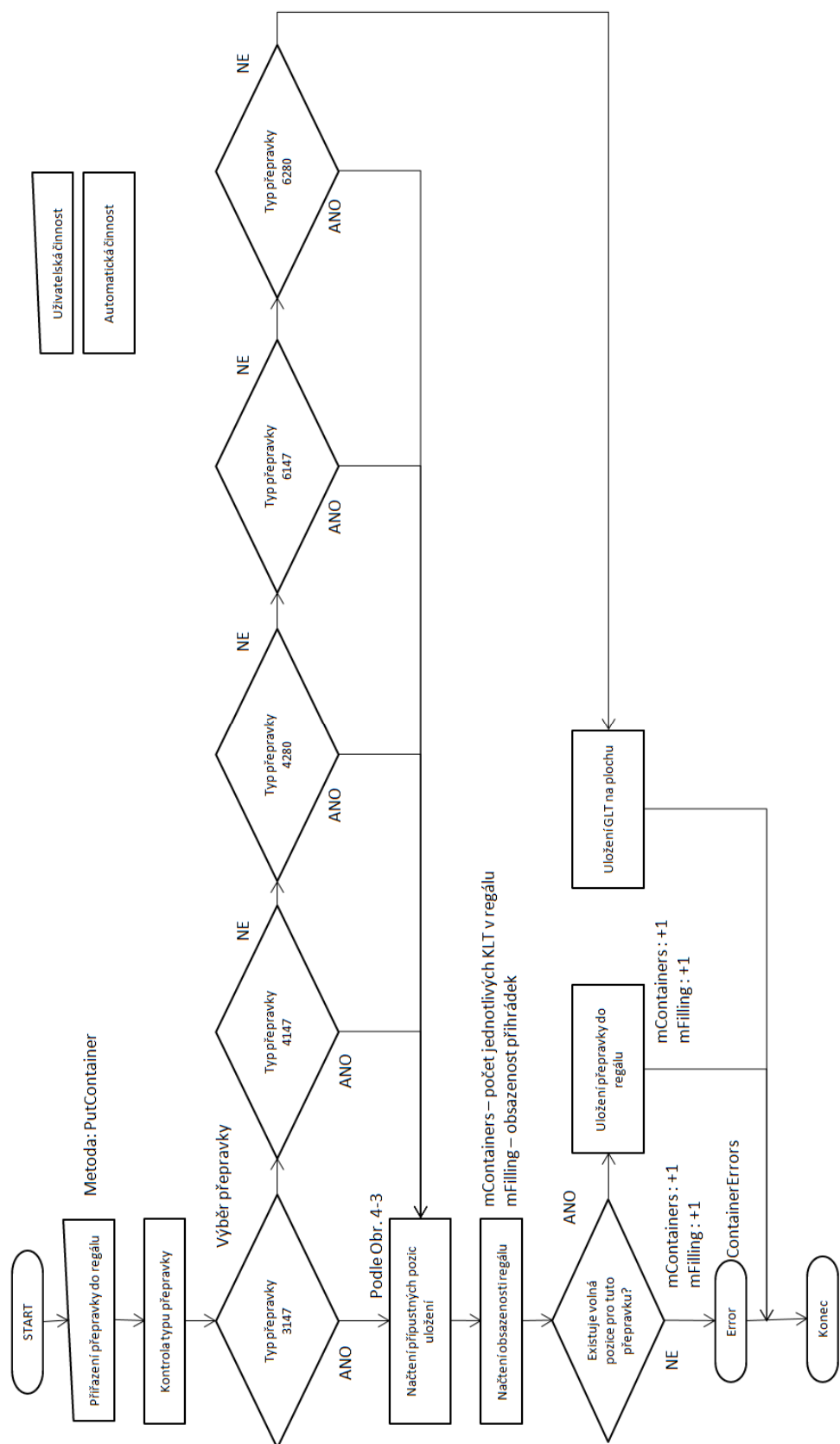
(autor)

a) prázdný regál

b) přeplněný regál

Dále je zde možnost nastavení atributu „zpětný skluz“, který označuje sklon regálových přihrádek konkrétního regálu.

Algoritmus pro výpočet kombinací volných pozic v regálu (Obr. 4-15)



Obr. 4-15: Schéma plnění regálu

(autor)

Tyto výpočty probíhají ve třídě, která obsahuje všechny potřebné atributy popisující celkový počet, obsazenost a blokování regálových přihrádek.

Tato třída má v sobě naprogramované metody zajišťující plnění regálu. To jsou metody pro vkládání a odebírání přepravek. Dále pak pro kompletní přepočítávání aktuálního stavu obsazenosti a vyhodnocení přeplněnosti regálu.

Vkládání KLT do regálu je prováděno tímto způsobem

- Metoda spouští kontrolu vkládané přepravky.
- Pokud typ přepravky vyhovuje, vloží se dovnitř.

Následně se spustí metoda pro přepočítání obsazenosti regálu. Načte se současně naplnění regálu a následně se v cyklu prochází celé pole přípustných přihrádek.

- a) Pokud je na pozici volno, inkrementuje se obsazení přihrádky regálu o 1.
- b) Pokud je již obsazeno, zkouší se naplnit další přípustná pozice.
- c) Pokud je již všude obsazeno (tzn. že přepravka se nikam nedá umístit), nastaví se indikace přeplnění regálu. Zároveň se do pole s chybami přihrádek zaznamená, že je s touto přepravkou problém. Tímto dojde k přeplnění regálu a obsazení se inkrementuje o 1.

Pole s chybami přihrádek se dále používá pro zvýraznění pozic v dialogu, kde došlo k chybě. Indexy tohoto pole KLT-přepravek označují druh chyby (přeplněný regál, nulová kapacita pro přepravku, ...)

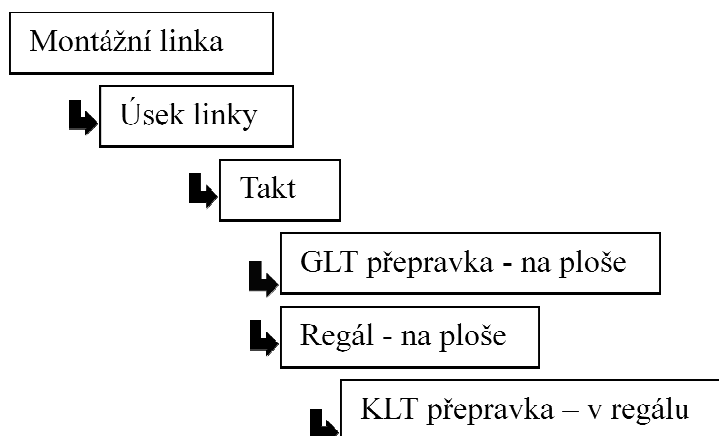
4.3.3. Skript zapisující data do DPE

Vytvoří v DPE novou logistickou strukturu s úložišti.

Tento skript je kvůli jednoduchosti výsledně obsažen ve skriptech „MaterialPlanning“ a „MaterialPlanning_engine“.

Přebírá aktuální data z paměti DataTable aplikace a generuje novou strukturu ve stromě logistických úložišť. To znamená, že pokud již nějaká struktura existovala, tak ji smaže a následně vytvoří novou podle aktuálních dat.

Skript prochází celý seznam a vytváří strukturu úložišť v DPE (Obr. 4-16).



Obr. 4-16: Nová generovaná struktura úložišť

(autor)

Struktura je tvořena tak, že se převezme konkrétní takt na úseku linky. Vytvoří se uzel „takt“ s jeho označením. Pro něj jsou vytvořeny objekty regálů, které se následně vytvoří, jako nižší uzly. Pokud se na tomto taktu vyskytují i přepravky GLT, najdou se v knihovně DPE a navážou se přímo na uzel „takt“. Pro KLT přepravky se najde správný regál, pro který se provede navázání vyhledané přepravky v knihovně DPE.

Protože může být generování této struktury zdlouhavý proces, bylo použito okno s komponentou progressbar. Během tohoto procesu je spuštěn v popředí, aby dotyčný uživatel byl seznámen s tím, že systém něco provádí.

4.3.4. Implementace do systému

Nejdříve je nutné změnit ve vývojovém prostředí původní vytvářený typ aplikace z Windows form application na ClassLibrary. Tím vznikne projekt, který má jako výstupní zkompileovaný soubor knihovnu „dll“.

Po této transformaci aplikace na knihovnu je již možné ji zaregistrovat do systému.

Registrace knihoven se realizuje tímto způsobem:

- Nejprve je nutné v příkazovém řádku namapovat správné úložiště vytvořené knihovny.
- Dále je možné spustit registrační soubor pro knihovny „regasm.exe“.
(C:\Windows\Microsoft.NET\Framework64\v4.0.30319\regasm.exe
název_knihovny.dll)

Při dodržení správného postupu, dojde k úspěšné registraci knihovny do systému. Po oznámení správné registrace je již možné tuto knihovnu volat.

Pro spouštění již implementované aplikace je nutné mít založený skript, který volá objekt z příslušné knihovny (Náhled č. 4-5). Po vytvoření tohoto objektu je již možné zobrazit dialogové okno, vytvářené právě pro tuto implementovanou aplikaci. Po zavolání se již automaticky spustí aplikace pro návrh inteligentních regálů jako okno, které náleží systému DPE.

```
Sub Main
    set RackForm = CreateObject ("Placpart.Core.Form1")
    call RackForm.showDialog()
End Sub
```

Náhled č. 4-1: Skript v DPE odkazující na registrovanou knihovnu dll

4.4. Materiály pro školení

Pro seznámení uživatelů s touto novou funkcí byla vytvořena prezentace, popisující všechny funkce a postupy, které tato úprava přináší. Výběr hlavních částí z této prezentace je uveden v příloze č. 12-17.

4.5. Testování

Testování proběhlo podle již zmiňovaných pravidel. Pro zjištění funkčnosti výše zmiňovaných úprav byli vybráni 3 uživatelé. Každý z nich měl za úkol projít všechny body ve svém protokolu.

Celý test byl prováděn na testovacím serveru u zákazníka a probíhal přibližně 6 hodin. Bylo odzkoušeno mnohem více prováděných úkonů, než bylo předem připraveno. Během zkoušení těchto funkcí se nic problematického nevyskytlo. Pouze bylo navrženo od uživatelů několik úprav, které by se mohli provést. Tyto úpravy by zjednodušily a urychlily orientaci v zobrazovací tabulce. Z funkčního hlediska bylo tedy vše v pořádku, a tudíž bylo možné tyto změny vyhodnotit jako funkční a aplikovat je i na produkční server.

4.6. Navázání na tuto úlohu

Po splnění všech částí této úlohy, je možné navázat na tuto problematiku automatickým generováním regálů v 3D layoutech montážní linky, podle přesných údajů v DPE. V této nové struktuře bude již možné generovat layout montážní linky včetně předem definovaných parametrů regálů. Tyto regály vyobrazené ve 3D prostředí DPE mohou nést svá označení i označení dílů, kterých se v nich budou při reálné montáži vyskytovat. Po zobrazení 3D layoutu bude možné na regálu přečíst, které díly budou v konkrétním regálu uloženy. Dále tyto informace budou sloužit pro usnadnění výstavby pracovišť na celé montážní lince.

Závěr

Cílem diplomové práce bylo rozšíření softwaru Delmia Process Engineer o funkcionality, které podstatně zjednoduší plánování rozmístění materiálu na montážní lince. Tím vznikl zcela samostatný modul, integrovaný do datového prostředí DPE.

Bylo nutné se podrobně seznámit s architekturou DPE, s možnostmi jeho úprav a rozšiřování. Bylo důležité správně pochopit plánovací a výrobní procesy společnosti, používající daný software.

Je zde popsána digitální továrna a její přínosy v současném výrobním průmyslu. Pro objasnění potřebných pojmů obsahuje práce informace o základních nástrojích, používaných pro digitální továrnu. Nástroj, jenž byl předmětem úprav, je popsán podrobněji. Jelikož byla tato práce zaměřena především na úpravy tohoto systému, byly zde popsány způsoby, jak při tomto rozšiřování postupovat. Na základě těchto informací bylo možné postupovat dále a zabývat se problémy, které bylo možné pomocí těchto způsobů úprav vyřešit.

Tyto problémy se týkaly zjednodušení a vylepšení konkrétních úkonů uživatelů DPE u zákazníka. U těchto úloh bylo nutné prokonzultovat veškeré podrobnosti s uživateli a následně bylo možné přesně definovat zadání úloh. Pro tyto úlohy byly vytvořeny návrhy řešení. Po opětovné konzultaci s uživateli byly požadavky upřesněny. Následně bylo již možné navrhnout výsledné řešení, podle kterého byly úlohy realizovány.

První úloha umožní technologům bez jakýchkoli zásahů přiřazovat provádění operací k jednotlivým obdobím a následně se těmito vlastnostmi již nezabývat. Uživatel se může spolehnout na automatické aktualizování provádění operací a nemusí se již vracet k předchozím nastavením a upravovat tyto vlastnosti operace manuálně. Úprava byla řešena dvěma skripty, přičemž jeden byl jednorázový skriptový příkaz, zaměřený na úpravu dat v celé databázi. Druhý byl nastaven jako skriptová akce, provádějící úpravu dat na jednom objektu v závislosti na úkonu uživatele.

Druhá úloha umožní uživatelům jednoduchou a přehlednou práci s přiřazováním přepravků do regálů na konkrétních úsecích výrobní linky. Místo původně používané aplikace, která měla velmi opožděně aktualizovaná data, se vytvořila úplně nová

funkcionalita DPE. Ta umožňuje používat všechny funkce, které měl uživatel původně k dispozici, ale s tím rozdílem, že může konečně pracovat s aktuálními daty.

Úprava byla uskutečněna pomocí skriptů, pracujících s daty v DPE, a vývojem nové aplikace, která byla následně implementována do systému.

Pro uživatele byla vytvořena prezentace, ve které byli seznámeni se všemi funkcemi a postupy, které tato úprava přináší. Tato prezentace je použita jako návod pro používání této nové funkcionality v DPE.

K této úloze bylo navrženo pokračování, o které by se mohlo DPE rozšířit. Jde o automatické generování 3D layoutu, zobrazující všechna úložiště ve 3D prostředí.

Veškerá rozšíření, která byla zákazníkem požadována, byla splněna. Vše bylo následně úspěšně otestováno na testovacím serveru. Po ověření funkcí byly změny uvedeny do provozu i na produkčním serveru. Uživatelé, kteří dostali tyto úpravy k dispozici, byli spokojeni jak z pohledu funkcí, tak i přehlednosti ovládání.

Použitá Literatura

[1] *User Manual DELMIA Process Engineer*. verze: 5.18: Dassault Systemes, 2007.

[2] Mareček, Petr. Digitální továrna DELMIA. 2009. 31s. prezentační materiály T-systems.

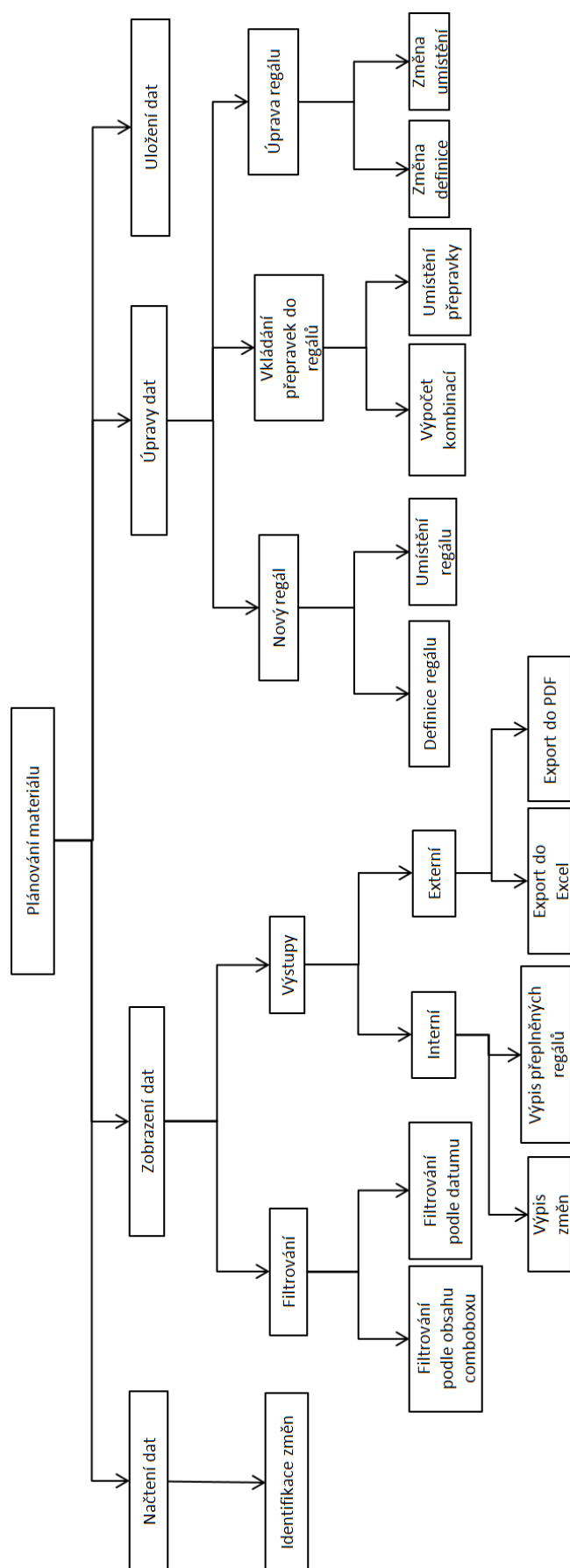
[3] Vývojářské fórum o jazyku Visual Basic .NET. 2012. Vývojářské materiály. Dostupné z WWW: <[http:// http://www.vbnet.cz/](http://http://www.vbnet.cz/)>.

PŘÍLOHY

Seznam příloh

Příloha č. 1 – Funkční diagram aplikace pro aplikaci materiálového plánování.....	65
Příloha č. 2 - Knihovna s 3D komponentami	66
Příloha č. 3 – Visual Studio 2010 - nastavení tvorby knihovny dll obr. 1	66
Příloha č. 4 - Visual Studio 2010 - nastavení tvorby knihovny dll obr. 2	67
Příloha č. 5 – Menu aplikace 1	67
Příloha č. 6 - Menu aplikace 2	68
Příloha č. 7 - Menu aplikace 3	68
Příloha č. 8 – Okno změny typu balení.....	68
Příloha č. 9 – Hromadná změna úložiště	69
Příloha č. 10 – Změna úložiště – výběr cílového umístění	69
Příloha č. 11 – Stručné informace o regálu.....	69
Příloha č. 12 – Prezentace nové funkce obr. 1.....	70
Příloha č. 13 – Prezentace nové funkce obr. 2.....	70
Příloha č. 14 – Prezentace nové funkce obr. 3.....	71
Příloha č. 15 – Prezentace nové funkce obr. 4.....	71
Příloha č. 16 – Prezentace nové funkce obr. 5.....	72
Příloha č. 17 - Prezentace nové funkce obr. 6	72
Příloha č. 18 - Prezentace nové funkce obr. 7	73
Příloha č. 19 - Prezentace nové funkce obr. 8	73
Příloha č. 20 - Prezentace nové funkce obr. 9	74
Příloha č. 21 - Prezentace nové funkce obr. 10	74
Příloha č. 22 - Prezentace nové funkce obr. 11	75

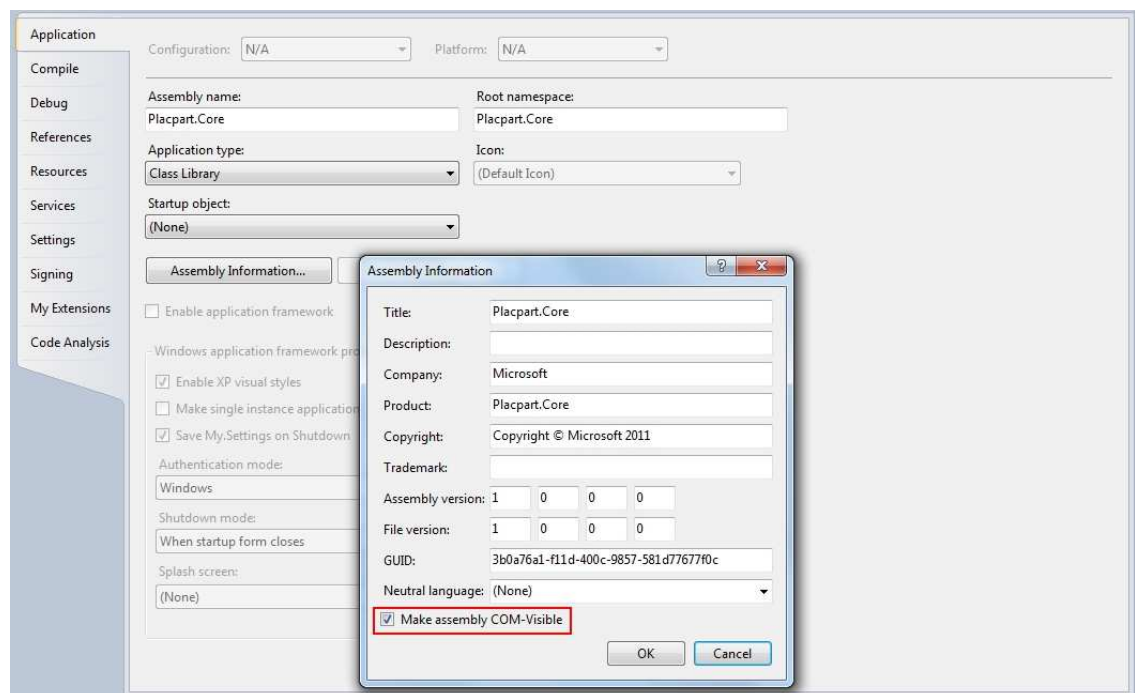
Příloha č. 1 – Funkční diagram aplikace pro aplikaci materiálového plánování



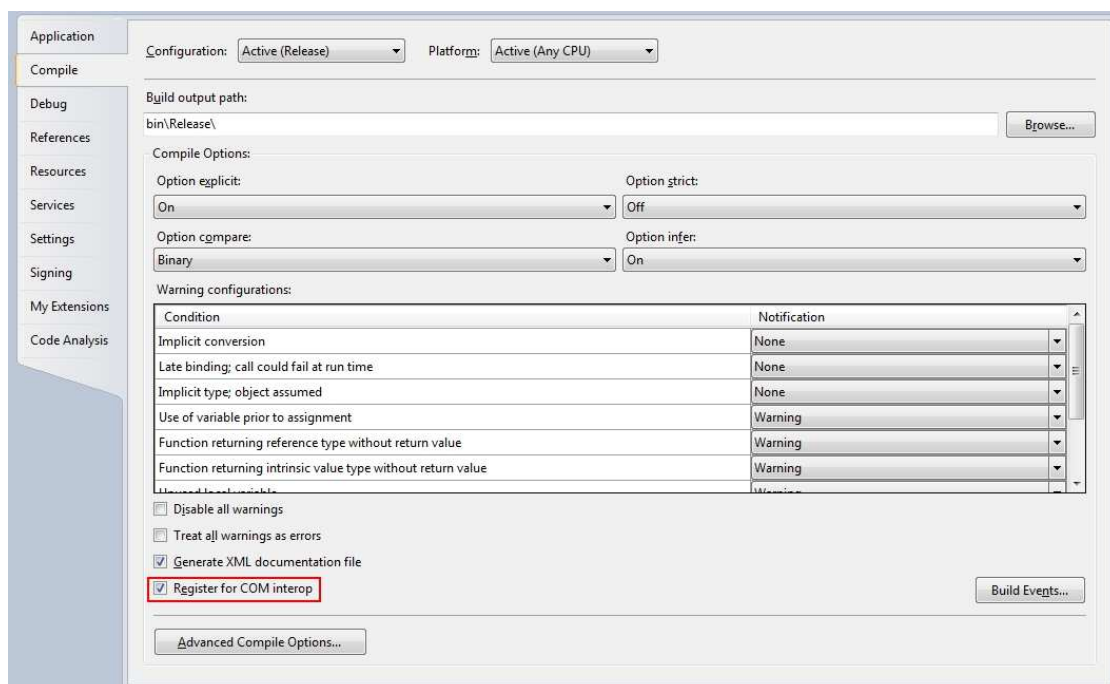
Příloha č. 2 - Knihovna s 3D komponentami

Work System Components		Obecné		Durchlaufregale	
		Systemelementnummer	Implicit Filter Behavior	Název	
WSC		__TEST_ZD	Normal Filtering	New Durchlaufregale	
Layout - Prvky		__Test_ZD_2	Normal Filtering	__Test_ZD_2	
Převraky - Speciál		0000001_Test	Normal Filtering	TEST_Duchlaufregal	
Převraky - Standard		00002_Test2	Normal Filtering	Test2	
Prvek elektroinstalace		001	Normal Filtering	Regal Skoda st	
Výrobní prostředek		141420	Normal Filtering	PP Standardní regál 01	
Stroje		1GF	Normal Filtering	Schwerlastregal gerade	
MGE Profil		234234	Normal Filtering	Faecherregal 1470x1200x1500	
Převraky na díly		281420	Normal Filtering	PP Standardní regál 02	
Pracovní otočné židle		2GF	Normal Filtering	Schwerlastregal schräg	
Pneumatické stolní lisy		421420	Normal Filtering	PP Standardní regál 03	
Standardní systémové prvky		561420	Normal Filtering	PP Standardní regál 04	
Nástroje		60010	Normal Filtering	Durchlaufregal Typ 1.1 A	
Osvětlení na pracovišti		60020	Normal Filtering	Durchlaufregal Typ 1.2 A	
Pracovní stoly		60030	Normal Filtering	Durchlaufregal Typ 2.1 A	
APS Gruppenecken		60040	Normal Filtering	Durchlaufregal Typ 2.2 A	
Automatisierte Bereiche		60050	Normal Filtering	Durchlaufregal Typ 3.1 A	
Befüllanlagen		60055	Normal Filtering	Durchlaufregal Typ 3.1 A - Sonderbreite	
Beleuchtung		60060	Normal Filtering	Durchlaufregal Typ 3.2 A	
Durchlaufregale		60070	Normal Filtering	Durchlaufregal Typ 4.1 A	
Energietechnik					
Fahrradständer					

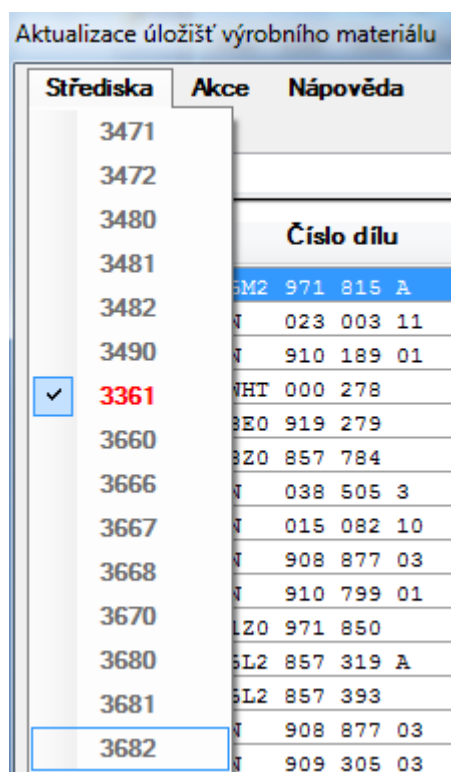
Příloha č. 3 – Visual Studio 2010 - nastavení tvorby knihovny dll obr. 1



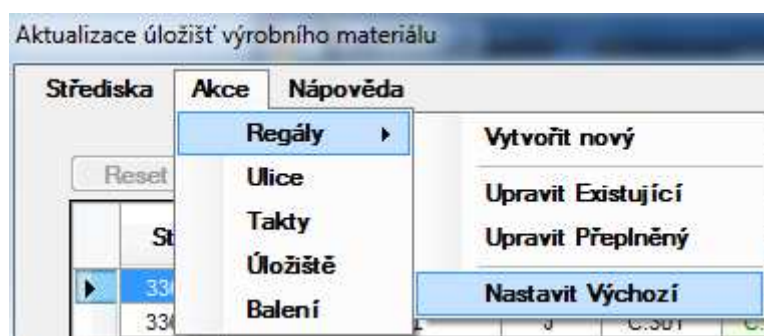
Příloha č. 4 - Visual Studio 2010 - nastavení tvorby knihovny dll obr. 2



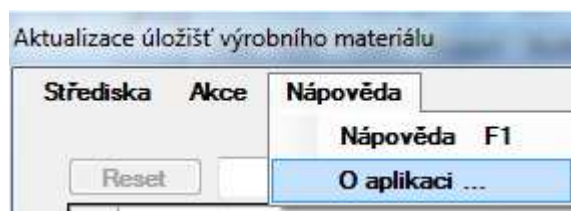
Příloha č. 5 – Menu aplikace 1



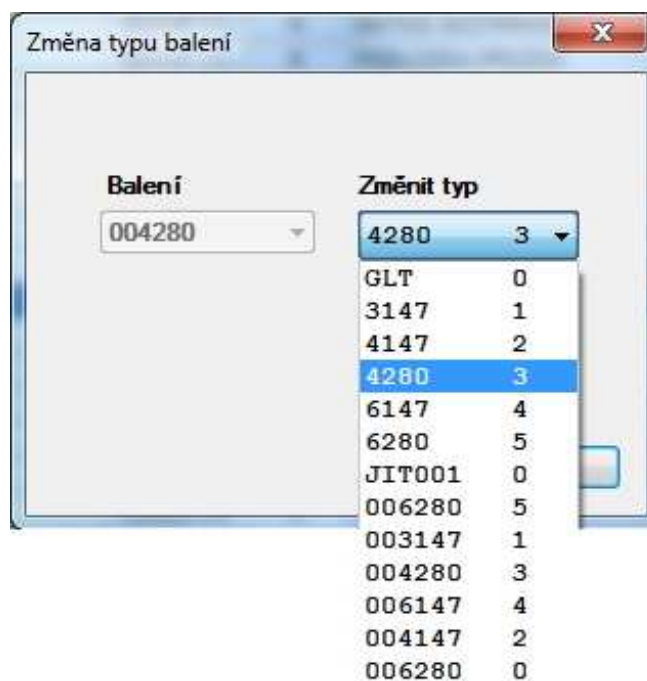
Příloha č. 6 - Menu aplikace 2



Příloha č. 7 - Menu aplikace 3



Příloha č. 8 – Okno změny typu balení



Příloha č. 9 – Hromadná změna úložiště

J	G.209	G.209.1		4147	N	DRZAK NADELU	N
J	G.209	G.209.1		4147	K	MATICE VKLADACI	N
J	G.209	G.209.1		3147	K	6-TI HRAN.SROUB SPL.HL.VNITR.	N
S	G.210	G.210		JIT001 (0)	S	VYPLN DVERI UPLNA	N
	G.210	G.210.1		6147		SKLO ODRAZOVE	
	G.210	G.210.1				SPINAC,ZRCATKA VNE	
	G.210	G.210.1				SVETLO DVERIVAROVNE	
	G.210	G.210.1				SKLO ODRAZOVE	
	G.210	G.210.1		3147		HR	
	G.210	G.210.1		3147		SROUB	
	G.210	G.210.1		3147		SROUB	
	G.211	G.211.1		6280		KLIKA SPOUST.SKLA	

Změnit Úložiště

Počet v regálu

Hromadná změna úložiště

Příloha č. 10 – Změna úložiště – výběr cílového umístění

Výběr cílového regálu

Středisko

Ulice

Takt

Č.regálu

3361

G

211

1

3361 - G.211.1

Příloha č. 11 – Stručné informace o regálu

WHT 000 278	J	C.301	C.301.2		003147 (1)
8E0 919 279	J	D.301	D.301.1		004147 (2)
8Z0 857 784	J	D.301	D.301.1		003147 (1)
N 038 505 3	J	D.301	D.301.1		003147 (1)

Regál
Kapacita: 0.8.0.0.6
Naplněno: 2.1.0.0.0

Layout2 – Seznámení s aplikací

Obsah

1. Příprava struktury
 - 1.1. Příprava úseků Layoutu
2. Spuštění aplikace
3. Správa změn - přehled
 - 3.1. Tabulka změn
 - 3.2. Razení
 - 3.3. Filtrování
4. Úpravy
 - 4.1. Zavedení balení
 - 4.2. Zavedení úložiště (Takt, Regál)
 - 4.3. Potvrzení změny (S,N,Z)
 - 4.3.1. Potvrzení více změn najednou
 - 4.4. Úpravy regálu
 - 4.4.1. Přeplněný regál
 - 4.5. Změna úložiště
5. Ukončení aplikace
6. Aktualizace struktury Layoutu
 - 6.1. Průběh
 - 6.2. Výsledek
7. Závěr

Strana 1

Layout2 - Seznámení s aplikací

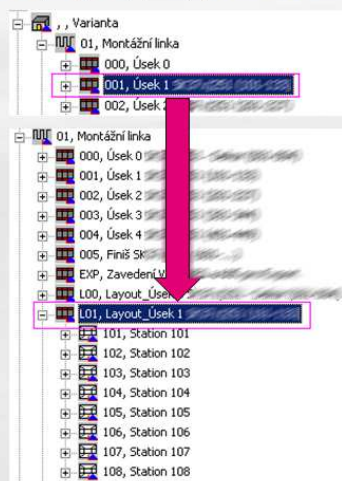
1. Příprava struktury

1.1. Příprava úseků Layoutu

Strukturu úseků Layoutu včetně jednotlivých taktů vytvoří aplikace sama, na základě Hrubého taktování (taktovací struktury ALB). Je však zatím nutné doplnit „ručně“ HLS layouty dané haly, případně přidat další 3D objekty (schodiště, sloup, rozvaděče, ohrady, stoly,...).

Úzly úseků včetně výše uvedených 3D objektů je také možné připravit předem. Aplikace pak vytvoří pouze [Takt,] Regály a Balení.

Nutno doladit standardizaci formátu HLS layoutu (rozměry a pozice) a jeho aktualizaci v DPE.



Úseky Layoutu vytváří aplikace takto:

- **Číslo úseku** je třímístné a je složeno z prefixu „L“ a čísla původního ALB úseku (čili např. "001" => "L01")
- **Název úseku** je složeno z prefixu „Layout_“ a původního názvu úseku
- Prefixy se dají změnit podle požadavku...

(v dalších screenshotech se může objevit prefix „Layout_DPE_“, ...původně tam byl totiž „Layout_Placpart“, takže to bylo takové prvotní rozlišení... ale jak říkám, dá se to libovolně změnit...)

Strana 2

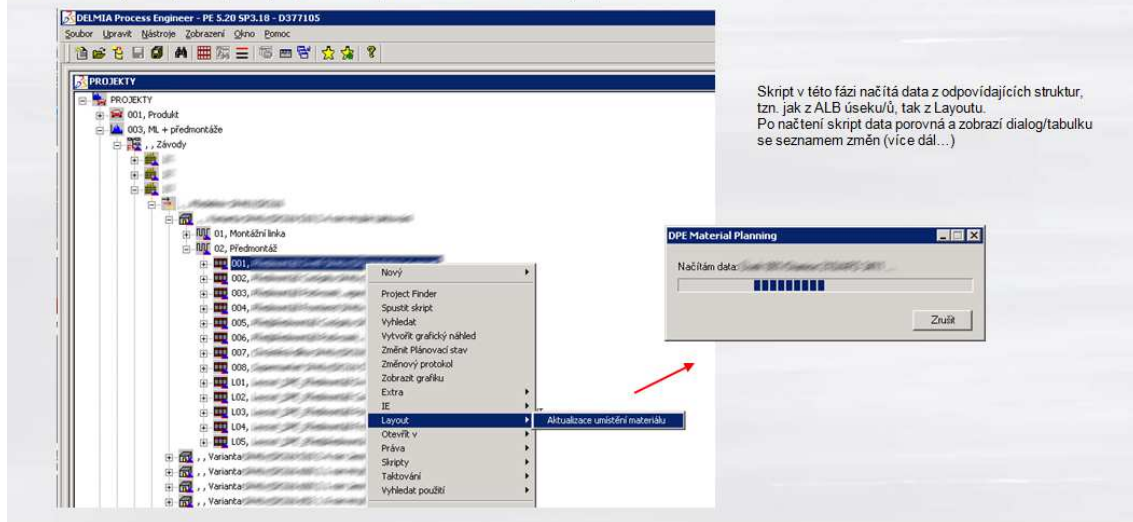
Layout2 - Seznámení s aplikací

2. Spuštění aplikace

2.1. Spuštění aplikace

Sériový technolog spustí aplikaci /skript pro změnový management na uzlu Haly, Montážní linky nebo úseku přes kontextové menu: **Layout / Aktualizace umístění materiálu** (dá se změnit).

Zobrazí se dialog hlásící průběh probíhající operace a skript začne načítat data ze struktury. Operaci lze zrušit.



Strana 3

Layout2 - Seznámení s aplikací

3. Správa změn

Menu

Filtiry – Textová pole: možno zadat i část hledaného textu
Rozbalovací seznamy: obsahují všechny právě zobrazené položky
 Filtiry lze libovolně kombinovat.

Názvy sloupců s možností řazení (přes klik)

Editovatelné sloupce

Hlavní tlačítka
Uložit – ukládá změny, provedené od spuštění aplikace, od posledního uložení nebo zrušení změn. Do provedení první úpravy je tlačítko nedostupné.
Zavřít / Zrušit – do provedení první změny/úpravy má tlačítko funkci zavření okna aplikace. Po provedení první změny se text změní „Zrušit“, a tlačítko má funkci zrušení změn, provedených od posledního uložení nebo Zrušení.

Barevné zvýraznění položek
 Zelená:
 - **Uloženo** – nová, nezavedená položka (nový regál/takt/úlice)
 - **S/N** (status) – indikace potvrzení změny
 Červená – signalizuje, že je něco v nepořádku:
 - **Stř.** (nákl. středisko) – neznámé nákladové středisko
 - **Uloženo** – přeplněný regál
 - **Balení** – nové nezavedené číslo balení

Stř.	Číslo dílu	W/H	Montáž	Uloženo	Poč.	Balení	Původ	Název dílu	S/N	Datum změny	Typ	Oper.	Sekv.
3361	370 837 085 A	J	G.200	G.200	111902 (?)	K	BOWDEN	TRMEN LOZISKA	N	09.03.2012	B6	3018	
3361	370 839 085 C	J	G.200	G.200	111902 (?)	K	BOWDEN	TRMEN LOZISKA	N		B6	3017	
3361	379 839 702	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		B6	3024	
3361	510 837 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 837 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)	K	BOWDEN	TESNENÍ DVERÍ VNEJ	N		SUV	3018	
3361	510 839 702 A	J	G.200	G.200	114888 (?)	K	TESNENÍ DVERÍ VNEJ	PLKT. 6 (?)	N		SUV	3024	
3361	510 839 085 A	J	G.200	G.200	114888 (?)								

Layout2 - Seznámení s aplikací

3.3 Správa změn - Filtrování

3.3. Filtrování

Textová pole:
- zadáním textu a potvrzením Enter se data vyfiltrují
- je možné zadat i část textu

Rozbalovací seznamy (comboboxy):
- k vyfiltrování dojde po výběru hodnoty ze seznamu

Aktivní filtr je barevně zvýrazněn

Filtry lze vzájemně kombinovat

Stř.	Číslo dílu	W/H	Montáž	Úložisko	Poč.	Bale
3361	319 839 649 A	J	C.206	C.206	5222	
3361	319 839 475	J	C.209	C.209	1233	
3361	319 839 477	J	C.209	C.209	1233	
3361	319 839 702	J	G.200	G.200	1148	
3361	319 839 904	J	G.205	G.205	5222	
3361	319 839 644 A	J	G.206	G.206	5222	
3361	319 839 476	J	G.209	G.209	1233	
3361	319 839 478	J	G.209	G.209	123315 (?)	K TESNENÍ STRAČI SK(SKLA)
3361	319 839 701	J	H.200	H.200	114888 (?)	K TESNENÍ DVERÍ VNEJ
3361	319 839 903	J	H.205	H.205	522291 (?)	K CLONA

Oper.	Sekv.
3195	
3243	
3243	
3024	
3193	
3194	
3242	
3242	
3023	
3192	

Strana 7

Layout2 - Seznámení s aplikací

4. Správa změn – Úpravy

4.1. Zavedení balení

Na začátku má aplikace definovaný výchozí seznam známých balení.
Seznam obsahuje jedno balení GLT (kód 0) a 5 typů KLT (kódy 1 až 5).
Ostatní balení jsou zatím **neznámá** a v sloupci Balení se zobrazují červeně (ve tvaru „číslo_balení (?)“). Těmto balením se dají přiřadit kódy z výchozího seznamu balení.

Zavedení balení zahájíte klikem pravého tlačítka myši na požadovaném čísle balení ve sloupci Balení. Objeví se dialog pro přiřazení kódu balení. Po přiřazení kódu se balení stane známým, zruší se u všech položek s tímto balením se zruší zvýraznění a v závorce za číslem balení bude přiřazený kód.

Stav po zavedení balení PKLT_6

Balení	Původ	Název dílu	S/N	Datum změny	Typ	Oper.	Sekv.
111902 (?)	K	BOMDEN	N		B6	3018	
522219 (?)	H	TRMEN LOŽISKA	N		B6	3018	
111902 (?)	K	BOMDEN	N		B6	3017	
114888 (0)	K	TESNENÍ DVERÍ VNEJ	N		B6	3024	
PKLT_6 (?)	K	BOMDEN	N		B6	3017	
114888 (0)	K	TESNENÍ DVERÍ V	N		B6	3017	
522219 (?)	H	TRMEN LOŽISKA	N		B6	3017	
006280 (?)	S	ZAMEN DVERÍ	N		B6	3017	
x (?)	S	SKLO DVERÍ ZAC	N		B6	3017	
4147	K	CEP	N		B6	3016	
111902 (?)	K	BOMDEN	N		B6	3015	
114888 (0)	K	TESNENÍ DVERÍ V	N		B6	3023	
PKLT_6 (?)	K	BOMDEN	N		B6	3016	
114888 (0)	K	TESNENÍ DVERÍ VNEJ	N		B6	3023	
522219 (?)	H	TRMEN LOŽISKA	N		B6	3016	
PKLT_6 (?)	K	BOMDEN	N		B6	3015	
114888 (0)	K	TESNENÍ DVERÍ VNEJ	N		B6	3023	
522219 (?)	H	TRMEN LOŽISKA	N		B6	3015	

Počet záznamů: 150 134 nové 0 dom. 0 změny

Strana 9

Layout2 - Seznámení s aplikací

4. Správa změn – Úpravy

4.2. Zavedení úložiště (Takt, Regál)

Nezavedená úložiště se zobrazují zelenou barvou (možno změnit).

Úložiště lze zavést (zlegalizovat) kliknutím pravým tlačítkem myši na buňku ve sloupci Úložiště.

Správa změn vyžaduje potvrzení pro dosud nezavedené složky adresy, tzn. Ulice, Takt nebo číslo Regálu.
Pokud uživatel v některém z potvrzení zvolí Ne, akce zavedení Úložiště se kompletně zruší a nedojde k žádným změnám.

Změna zobrazení adres po potvrzení. Je zrušeno barevné zvýraznění.

Pokud je Úložištěm Regál, zobrazí se vzápětí dialog pro úpravu parametrů Nového regálu.
Dialog pro nový regál se objeví i při vytvoření regálu pomocí menu: Akce/Regály/Vytvořit Nový, nebo při změně úložiště, kdy uživatel zadá adresu dosud neexistujícího regálu.

Strana 10

Layout2 - Seznámení s aplikací

4. Správa změn – Úpravy

4.4. Úprava parametrů regálu

Úpravu parametrů regálu zahájíme klikem pravým tlačítkem myši na buňku ve sloupci Úložiště. Úpravu lze spustit z jakékoliv položky náležející danému regálu.

Pokud úložiště ještě není zavedené, proběhne nejprve zavedení úložiště (viz snímek 9, Zavedení úložiště).

Poté se zobrazí se dialog Aktualizace pozic: Regál ...

Pro podporu aktuálnosti v zobrazení obsazení regálu Správa změn zjišťuje, jestli pro regál existují nějaké nepotvrzené změny. Pokud ano, nabídne uživateli možnost všechny tyto změny hromadně Potvrdit.
Po odsouhlasení akce se do regálu:
- vloží všechny Nové přepravy a
- vyjme všechny Stornované přepravy
Potvrzení bude probíhat podobně jako při akci Potvrzení změny (viz. Snímek 10, kap. 4.2. Potvrzení změny)

Popis dialogu Aktualizace pozic: Regál ...

Pozice: označení KLT / pozice.
Kapacita: celkové počty alokovaných přihrádek/pozic pro každou velikost přihrádky.
Blokováno: počty blokových pozic pro každou velikost přihrádky
Obsazeno KLT: počty „vložených“ KLT podle jejich typu/velikosti
Zpětný skluz: obsahuje regál zpětný skluz?

Dále dialog obsahuje informativní text s výčtem možných umístění jednotlivých typů KLT v jednotlivých typech přihrádek regálu.

Aktualizace úložiště výrobního materiálu

Pro tento regál existují nepotvrzené změny (9 nové, 0 storna).
Chcete potvrdit všechny změny?

Ano Ne

Strana 13

Layout2 - Seznámení s aplikací

4. Správa změn – Úpravy

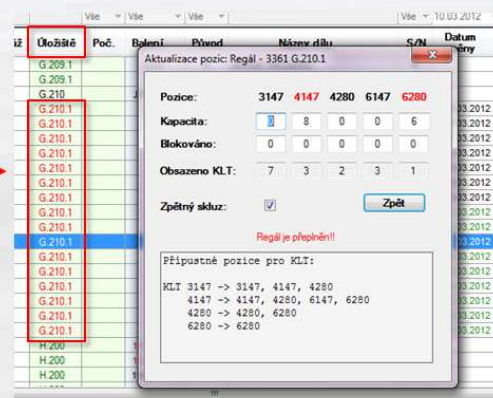
4.4.1 Úprava parametrů regálu – přeplněný regál

Přeplněný regál

Přeplnění regálu je indikováno:

- 1) V tabulce zvýrazněním textu **ve sloupci Úložiště**
- 2) V dialogu **Aktualizace pozic: Regál...** textem „Regál je přeplněný“ a zvýrazněním pozic/přehrádek, které jsou přeplněné.

Přeplnění regálu lze vyřešit změnou Kapacity a/nebo Blokování nebo v tabulce změn přesunout přepravky do jiného popř. nového regálu.



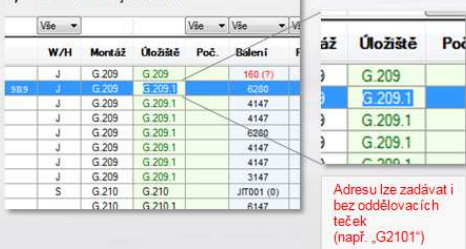
Strana 14

Layout2 - Seznámení s aplikací

4. Správa změn – Úpravy

4.5. Změna úložiště

1) Změna úložiště jednotlivě



2) Hromadná změna úložiště



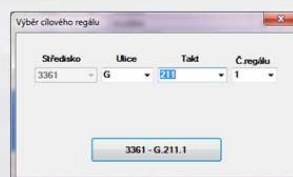
Změnu úložiště lze provést dvěma způsoby:

- 1) **Jednotlivě** - úpravou adresy **ve sloupci Úložiště**
Úpravu adresy zahájíte kliknutím do buňky na požadovaném řádku.
Pokud položka obsahuje nezavedené balení (?), zobrazí aplikace upozornění a dotaz na akci Zavedení balení.
Po zavedení balení je buňka s adresou připravena pro úpravu.
- 2) **Hromadně**, pro více položek naráz
V tabulce lze vybrat/označit požadované položky pomocí **Shift + normální klik** nebo **Ctrl + normální klik**, popř. jejich kombinací a pravým klikem na jednu z vybraných položek vyvolat kontextové menu. Pokud je změna úložiště možná*, bude v menu dostupný příkaz **Změnit úložiště**. Klikem na příkaz se vyvolá dialog pro výběr cílového úložiště (pro KLT to bude regál, pro GLT takt).

* Hromadná změna úložiště je možná, pokud jsou všechna balení ve výběru Zavedená a pokud jsou všechna vybraná balení buď KLT nebo GLT.

Pokud cílové úložiště ještě není zavedeno, proběhne nejprve akce Zavedení úložiště, popsána na snímku 9.

Pokud se cílový regál přeplní, aplikace se dotáže na spuštění úprav regálu.



Strana 15

Layout2 - Seznámení s aplikací

5. Správa změn - Ukončení

5.1. Aplikaci nelze opustit bez uložení nebo zrušení změn

